Kernelized Constrained Gaussian Fields and Harmonic Functions for Semi-supervised Learning

Celso A. R. Sousa
University of Fortaleza, Brazil
email: celso.sousa@unifor.br

Abstract—**Graph-based semi-supervised learning (SSL) methods are effective on many application domains. Despite such an effectiveness, many of these methods are transductive in nature, being uncapable to provide generalization for the entire sample space. In this paper, we generalize three existing graph-based transductive methods through kernel expansions on reproducing kernel Hilbert spaces. In addition, our methods can easily generate an inductive model in a parameter-free way, given a graph Laplacian constructed from both labeled and unlabeled examples and a label matrix. Through experiments on benchmark data sets, we show that the proposed methods are effective on inductive SSL tasks in comparison to manifold regularization methods.**

I. Introduction

Semi-supervised learning (SSL) aims to use the abundance of unlabeled examples to improve the classification performance in comparison to purely supervised learning methods. In scenarios in which we deal with only a few labeled examples, SSL methods can be an effective alternative. Among these methods, the graph-based ones are widely used due to their effectiveness on benchmark data sets [1]–[5] and great theoretical properties [6]–[9]. However, many of these methods were specifically designed for *transductive learning* tasks [10], being uncapable to provide generalization for the entire sample space.

Since many real applications need a model generated from training sample, it is crucial to provide novel inductive methods for graph-based SSL. In [11], the authors proposed an effective function induction for graph-based SSL and provided a linear time algorithm that is trained using a small subset of the examples. In [12], the authors proposed a technique for extending empirical functions based on the Nyström method [13], requiring the construction of a family of functions called geometric harmonics.

Unfortunately, the task of inductive graph-based SSL from transductive methods has taken little attention [14], [15]. In this paper, we generalize the *Gaussian Fields and Harmonic Functions* (GFHF) [16], *Robust Multi-class Graph Transduction* (RMGT) [17], and *RMGT with Higher Order Regularization* (RMGTHOR) [18] algorithms for inductive SSL tasks. We also generalize the method in [14] for combinatorial and normalized Laplacians.

A. Motivation

The motivations of this paper are summarized as follows:

- the method in [14] was designed for the combinatorial Laplacian. Since normalized Laplacians can give better results than combinatorial Laplacians [19], we can achieve better results by generalizing the method in [14] for normalized Laplacians;
- since graph-based SSL may be strongly dependent of parameter selection [20], there is a need of novel methods with fewer parameters, ideally none. The proposed methods in this paper are parameter-free, given a graph Laplacian generated from both labeled and unlabeled examples;
- despite the effectiveness of RMGTHOR, RMGT, and GFHF [20], [18], [21], these methods are naturally transductive. By generalizing these methods through kernel expansions, we can maintain their effectiveness on the unlabeled examples and also provide generalization for the whole input space;

B. Contributions

The contributions of this paper are summarized as follows:

- we generalize the method in [14] for combinatorial and normalized Laplacians. Specifically, we generate a method based on GFHF and approximate its closed-form solution for inductive tasks;
- we generalize GFHF through kernel expansions on *reproducing kernel Hilbert spaces* (RKHS), as in [22]. The generalized method yields the same output of GFHF for transductive learning tasks, independently of the kernel function used. We call the generalized method *Kernelized GFHF* (K-GFHF);
- we generalize RMGT and RMGTHOR using the same kernel expansions in [22]. The generalized method yields the same output of RMGTHOR and RMGT for transductive learning tasks, independently of the kernel function used. We call the generalized method *Kernelized Constrained GFHF* (K-CGFHF);
- we provide novel SSL methods with a few parameters. Given a graph Laplacian and a label matrix, K-GFHF and K-CGFHF generate a classification function in a parameter-free way for transductive learning tasks;

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

- we show that K-GFHF and K-CGFHF are effective on inductive SSL tasks using benchmarck data sets, achieving better results than the *manifold regularization* methods in [22].

C. Outline

The remainder of this paper is organized as follows. Section II provides a background on SSL. Section III provides heuristic methods for GFHF. Section IV describes the proposed methods. Section V shows our results on inductive SSL tasks. Finally, Section VI concludes the paper.

II. Background

This section revises GFHF, RMGT, and RMGTHOR. We also revise the process of kernel expansions on RKHS, as described in [22].

A. Notation

Consider a training sample $\mathcal{X} := \{\boldsymbol{x}_i\}_{i=1}^n \subset \mathbb{R}^d$, in which the first l examples are labeled with one of c classes while the remainder $u := n - l$ examples ($l+1 \le i \le n$) are unlabeled. Let $\mathbb{N}_a := \{i \in \mathbb{N}^* | 1 \le i \le a\}$, $\forall a \in \mathbb{N}^*$, and $\mathbb{B} := \{0,1\}$. Assume that $\boldsymbol{x}_i$, $i \in \mathbb{N}_l$, has label $y_i \in \mathbb{N}_c$. Let $\mathcal{N}_i \subset \mathcal{X}$ be the set of neighbors of $\boldsymbol{x}_i$. Let $\mathbf{Y}_\mathcal{L} \in \mathbb{B}^{l\times c}$ be a label matrix in which $(\mathbf{Y}_\mathcal{L})_{ij} = 1$ if and only if $\boldsymbol{x}_i$ has label $y_i = j$. Since we focus on multi-class problems, $\mathbf{Y}_\mathcal{L}\mathbf{1}_c = \mathbf{1}_l$ such that $\mathbf{1}_c$ is a c-dimensional 1-entry vector.

Let $\mathbf{A}_{i,\cdot} \in \mathbb{R}^{1\times b}$ and $\mathbf{A}_{\cdot,j} \in \mathbb{R}^a$ be the i-th row and j-th column vectors of a matrix $\mathbf{A} \in \mathbb{R}^{a\times b}$, $\forall a, b \in \mathbb{N}^*$. We write $\mathbf{A} \succcurlyeq 0$ to indicate that $\mathbf{A}$ is *positive semidefinite*[1] (PSD). The transpose of $\mathbf{A}$ is denoted by $\mathbf{A}^\top$. Consider $\mathbf{O}_{a\times b}$ the a-by-b null matrix and $\mathbf{I}_a$ the a-by-a identity matrix.

Let $\mathbf{W} \in \mathbb{R}^{n\times n}$ be a weighted matrix generated from $\mathcal{X}$. Assume that $\mathbf{F}$ and $\mathbf{Y}$ are subdivided into two submatrices while all other matrices are subdivided into four submatrices. For instance:

$$\mathbf{W} := \begin{bmatrix} \mathbf{W}_{\mathcal{L}\mathcal{L}} & \mathbf{W}_{\mathcal{L}\mathcal{U}} \\ \mathbf{W}_{\mathcal{U}\mathcal{L}} & \mathbf{W}_{\mathcal{U}\mathcal{U}} \end{bmatrix} \qquad \mathbf{F} := \begin{bmatrix} \mathbf{F}_\mathcal{L} \\ \mathbf{F}_\mathcal{U} \end{bmatrix}$$

where $\mathbf{W}_{\mathcal{L}\mathcal{L}} \in \mathbb{R}^{l\times l}$ and $\mathbf{F}_\mathcal{L} \in \mathbb{R}^{l\times c}$ are the submatrices of $\mathbf{W}$ and $\mathbf{F}$, respectively, on labeled examples, $\mathbf{W}_{\mathcal{U}\mathcal{U}} \in \mathbb{R}^{u\times u}$ and $\mathbf{F}_\mathcal{U} \in \mathbb{R}^{u\times c}$ are the submatrices of $\mathbf{W}$ and $\mathbf{F}$, respectively, on unlabeled examples, and so on.

Let $\mathbf{L} \in \mathbb{R}^{n\times n}$ be a *graph Laplacian* generated from $\mathbf{W}$. The *combinatorial* Laplacian $\mathbf{L}_\mathbb{C}$ is defined by $\mathbf{L}_\mathbb{C} := \mathbf{D} - \mathbf{W}$ where[2] $\mathbf{D} := \operatorname{diag}(\mathbf{W}\mathbf{1}_n)$. The *normalized* Laplacian $\mathbf{L}_\mathbb{N}$ is defined by $\mathbf{L}_\mathbb{N} := \mathbf{I}_n - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ where $\mathbf{I}_n$ is the n-by-n identity matrix. By construction, we have $\mathbf{L}_\mathbb{C} \succcurlyeq 0$ and $\mathbf{L}_\mathbb{N} \succcurlyeq 0$.

[1]A matrix $\mathbf{A} \in \mathbb{R}^{a\times a}$ is PSD if it is symmetric with nonnegative eigenvalues. If $\mathbf{A}$ is PSD, we have $\boldsymbol{v}^\top \mathbf{A} \boldsymbol{v} \ge 0$, $\forall \boldsymbol{v} \in \mathbb{R}^a$.

[2]Given a vector $\boldsymbol{v} \in \mathbb{R}^a$, the operation $\operatorname{diag}(\boldsymbol{v})$ outputs an $a \times a$ matrix $\mathbf{A}$ such that $\mathbf{A}_{ii} = \boldsymbol{v}_i$, $\forall i \in \mathbb{N}_a$.

B. GFHF

GFHF[3] [16] can be formulated as the following optimization problem:

$$\min_{\mathbf{F}\in\mathbb{R}^{n\times c}} \operatorname{tr}\left(\mathbf{F}^\top \mathbf{L}\mathbf{F}\right) \quad \text{s.t.} \quad \mathbf{F}_\mathcal{L} = \mathbf{Y}_\mathcal{L} \tag{1}$$

where $\operatorname{tr}(\mathbf{A}) := \sum_{i=1}^a \mathbf{A}_{ii}$ is the trace of a matrix $\mathbf{A} \in \mathbb{R}^{a\times a}$. Since $\mathbf{L} \succcurlyeq 0$, we obtain the following closed-form solution:

$$\mathbf{F} = \begin{bmatrix} \mathbf{Y}_\mathcal{L} \\ -\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{Y}_\mathcal{L} \end{bmatrix} \tag{2}$$

As proved in [14], when $\mathbf{L} = \mathbf{L}_\mathbb{C}$, $\mathbf{F}$ is always row-normalized, i.e., $\mathbf{F}\mathbf{1}_c = \mathbf{1}_n$. Due to the *harmonic property* [16], $\mathbf{F} \ge 0$. Therefore, $\mathbf{F}_{ij}$ acts like a posterior probability of $\boldsymbol{x}_i$ being of class j when $\mathbf{L} = \mathbf{L}_\mathbb{C}$.

The minimization problem in (1) can be iteratively solved for $\mathbf{L} = \mathbf{L}_\mathbb{C}$. Let $\mathbf{P} := \mathbf{D}^{-1}\mathbf{W} \in \mathbb{R}^{n\times n}$ be the probability transition matrix. Given an initial matrix $\mathbf{F}_\mathcal{U}^{(0)} \in \mathbb{R}^{u\times c}$, we iterate:

$$\mathbf{F}_\mathcal{U}^{(t+1)} = \mathbf{P}_{\mathcal{U}\mathcal{U}}\mathbf{F}_\mathcal{U}^{(t)} + \mathbf{P}_{\mathcal{U}\mathcal{L}}\mathbf{Y}_\mathcal{L} \tag{3}$$

This converges to:

$$\mathbf{F}_\mathcal{U}^* = \left(\mathbf{I}_u - \mathbf{P}_{\mathcal{U}\mathcal{U}}\right)^{-1}\mathbf{P}_{\mathcal{U}\mathcal{L}}\mathbf{Y}_\mathcal{L} = -(\mathbf{L}_\mathbb{C})_{\mathcal{U}\mathcal{U}}^{-1}(\mathbf{L}_\mathbb{C})_{\mathcal{U}\mathcal{L}}\mathbf{Y}_\mathcal{L}$$

C. RMGT and RMGTHOR

RMGT [17] can be viewed as the following constrained optimization problem:

$$\begin{gathered} \min_{\mathbf{F}\in\mathbb{R}^{n\times c}} \operatorname{tr}\left(\mathbf{F}^\top \mathbf{L}_\mathbb{C}\mathbf{F}\right) \\ \text{s.t.} \quad \mathbf{F}_\mathcal{L} = \mathbf{Y}_\mathcal{L}, \;\; \mathbf{F}\mathbf{1}_c = \mathbf{1}_n, \;\; \mathbf{F}^\top\mathbf{1}_n = n\boldsymbol{\omega} \end{gathered} \tag{4}$$

The closed-form solution of (4) is given by:

$$\mathbf{F} = \begin{bmatrix} \mathbf{Y}_\mathcal{L} \\ -(\mathbf{L}_\mathbb{C})_{\mathcal{U}\mathcal{U}}^{-1}(\mathbf{L}_\mathbb{C})_{\mathcal{U}\mathcal{L}}\mathbf{Y}_\mathcal{L} + \dfrac{(\mathbf{L}_\mathbb{C})_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u}{\mathbf{1}_u^\top(\mathbf{L}_\mathbb{C})_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u}\boldsymbol{\tau} \end{bmatrix} \tag{5}$$

in which

$$\boldsymbol{\tau} = n\boldsymbol{\omega}^\top - \mathbf{1}_l^\top\mathbf{Y}_\mathcal{L} + \mathbf{1}_u^\top(\mathbf{L}_\mathbb{C})_{\mathcal{U}\mathcal{U}}^{-1}(\mathbf{L}_\mathbb{C})_{\mathcal{U}\mathcal{L}}\mathbf{Y}_\mathcal{L}$$

RMGT was specifically designed for the combinatorial Laplacian $\mathbf{L}_\mathbb{C}$, which may not provide effective classification performances in real applications [19]. Such an issue was solved in [18], generating a method called RMGTHOR. Specifically, RMGTHOR [18] is a generalization of RMGT for any graph Laplacian. Solving (4) for a general Laplacian $\mathbf{L}$ yields the following closed-form solution:

[3]See [14], [15] for a review.

$$\mathbf{F} = \begin{bmatrix} \mathbf{Y}_{\mathcal{L}} \\ -\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}} + \frac{\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u}{\mathbf{1}_u^\top \mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u}\boldsymbol{\tau} + \frac{1}{c}\boldsymbol{\nu}\mathbf{1}_c^\top \end{bmatrix} \tag{6}$$

in which

$$\boldsymbol{\tau} = n\boldsymbol{\omega}^\top - \mathbf{1}_l^\top \mathbf{Y}_{\mathcal{L}} + \mathbf{1}_u^\top \mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}}$$
$$\boldsymbol{\nu} = \mathbf{1}_u + \mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{1}_l - \frac{\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u}{\mathbf{1}_u^\top \mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u}\left(u + \mathbf{1}_u^\top \mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{1}_l\right)$$

D. Kernel expansions on RKHS

Given a binary classification problem ($y_i \in \{-1, +1\}$, $\forall i \in \mathbb{N}_l$), the *manifold regularization* framework [22] can be viewed as the following optimization problem:

$$\min_{f \in \mathcal{H}_{\mathcal{K}}} \frac{1}{l}\sum_{i=1}^{l} \mathcal{V}(\boldsymbol{x}_i, y_i, f) + \gamma_A \|f\|_{\mathcal{H}_{\mathcal{K}}} + \gamma_I \boldsymbol{f}^\top \mathbf{L}\boldsymbol{f} \tag{7}$$

where $\mathcal{K}(\cdot,\cdot)$ is a kernel function and $\mathbf{K} \in \mathbb{R}^{n\times n}$ is a kernel matrix in which $\mathbf{K}_{ij} := \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\forall i, j \in \mathbb{N}_n$. $\mathcal{H}_{\mathcal{K}}$ is the *Reproducing Kernel Hilbert Space* (*RKHS*) for the kernel $\mathcal{K}$, $\boldsymbol{f} := [f(\boldsymbol{x}_1), \cdots, f(\boldsymbol{x}_n)]^\top \in \mathbb{R}^n$, $\mathcal{V}(\boldsymbol{x}_i, y_i, f)$ is a cost function, $\|\cdot\|_{\mathcal{H}_{\mathcal{K}}}$ is the norm in $\mathcal{H}_{\mathcal{K}}$, and γ_A and γ_I are the *ambient* and *intrinsic* regularization parameters, respectively.

Due to the *Representer Theorem* in [22], the solution of (7) can be written as $f(\boldsymbol{x}) = \sum_{i=1}^{n} \mathcal{K}(\boldsymbol{x}, \boldsymbol{x}_i)\boldsymbol{\alpha}_i$ with $\boldsymbol{\alpha} \in \mathbb{R}^n$. Therefore, we can classify the examples in the input space through kernel expansions over both labeled and unlabeled examples.

III. Heuristic induction for GFHF

This section generalizes the approach in [14], including special cases for $\mathbf{L} = \mathbf{L}_{\mathbb{C}}$ and $\mathbf{L} = \mathbf{L}_{\mathbb{N}}$.

A. Notation

We used the same notation in [14]. After training stage, we consider that the examples in $\mathcal{X}$ are labeled, i.e., each $\boldsymbol{x}_i \in \mathcal{X}$ has a corresponding "*label vector*"[4] $\mathbf{F}_{i,\cdot} \in \mathbb{R}^c$, which comes from the transductive solution $\mathbf{F}$. Let $\widetilde{\boldsymbol{x}} \in \mathbb{R}^d$ be a test example and $\widetilde{\mathcal{X}} := \mathcal{X} \cup \{\widetilde{\boldsymbol{x}}\} \subset \mathbb{R}^d$ be a sample in the input space. The main idea of the heuristic induction process is to run a transductive method from $\widetilde{\mathcal{X}}$ and provide an approximated solution for $\widetilde{\boldsymbol{x}}$ as fast as possible.

Let $\widetilde{\mathbf{W}} \in \mathbb{R}^{(n+1)\times(n+1)}$ be the weighted matrix associated to $\widetilde{\mathcal{X}}$. Since self-loops are not permitted, $\widetilde{\mathbf{W}}_{\mathcal{U}\mathcal{U}} = 0$. Considering $\widetilde{\mathcal{N}} \subset \mathcal{X}$ the set of neighbors of $\widetilde{\boldsymbol{x}}$, we have $\left(\widetilde{\mathbf{W}}_{\mathcal{U}\mathcal{L}}^\top\right)_i = \mathcal{K}(\widetilde{\boldsymbol{x}}, \boldsymbol{x}_i)$ if $\boldsymbol{x}_i \in \widetilde{\mathcal{N}}$ and 0 otherwise. In order to decrease the computation time required for induction, we consider that Assumption 1 holds[5]. Mathematically, such an assumption states that $\widetilde{\mathbf{W}}_{\mathcal{L}\mathcal{L}} = \mathbf{W}$.

[4]Now we have *soft* (real-valued) labels, instead of binary ones.

[5]If Assumption 1 does not hold, we have to reconstruct $\mathbf{W}$ for each test example $\widetilde{\boldsymbol{x}}$. This is uneffective even for a small training sample.

Assumption 1 (Graph connectivity for induction [15]): The graph connectivity on the training examples does not change during the induction process. Intuitively, we assume that $\mathbf{W}$ *is constant.*

Let $\widetilde{\mathbf{N}} \in \mathbb{B}^n$ be a neighborhood vector in which $\widetilde{\mathbf{N}}_i = 1$ if and only if $\boldsymbol{x}_i \in \widetilde{\mathcal{N}}$, $\Psi : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}_+$ a distance function, and $\widetilde{\boldsymbol{\Psi}} \in \mathbb{R}^n$ a distance vector in which $\widetilde{\boldsymbol{\Psi}}_i := \Psi(\widetilde{\boldsymbol{x}}, \boldsymbol{x}_i)$, $\forall i \in \mathbb{N}_n$. Consider $\widetilde{\mathbf{F}} \in \mathbb{R}^{(n+1)\times c}$ the output matrix of an SSL method for $\widetilde{\mathcal{X}}$ and $\widetilde{\mathbf{Y}}_{\mathcal{L}} := \mathbf{F} \in \mathbb{R}^{n\times c}$ the corresponding label matrix.

B. Inductive method

We call the generalized heuristic method *Heuristic GFHF* (H-GFHF). For induction, the optimization problem in (1) becomes:

$$\min_{\widetilde{\mathbf{F}} \in \mathbb{R}^{(n+1)\times c}} \operatorname{tr}\left(\widetilde{\mathbf{F}}^\top \widetilde{\mathbf{L}}\widetilde{\mathbf{F}}\right) \quad \text{s.t.} \quad \widetilde{\mathbf{F}}_{\mathcal{L}} = \mathbf{F} \tag{8}$$

This yields the following closed-form solution:

$$\widetilde{\mathbf{F}} = \begin{bmatrix} \mathbf{F} \\ -\widetilde{\mathbf{L}}_{\mathcal{U}\mathcal{U}}^{-1}\widetilde{\mathbf{L}}_{\mathcal{U}\mathcal{L}}\mathbf{F} \end{bmatrix} \tag{9}$$

Therefore, we classify $\widetilde{\boldsymbol{x}}$ as:

$$y\left(\widetilde{\boldsymbol{x}}\right) = \arg\max_{i \in \mathbb{N}_c} \left(\widetilde{\mathbf{F}}_{\mathcal{U}}\right)_i = \arg\max_{i \in \mathbb{N}_c} \left(-\widetilde{\mathbf{L}}_{\mathcal{U}\mathcal{U}}^{-1}\widetilde{\mathbf{L}}_{\mathcal{U}\mathcal{L}}\mathbf{F}\right)_i \tag{10}$$

Eq. (10) provides an inductive method for GFHF for a given graph Laplacian $\mathbf{L}$. In Propositions 1 and 2, we provide special cases of (10) for $\mathbf{L} = \mathbf{L}_{\mathbb{C}}$ and $\mathbf{L} = \mathbf{L}_{\mathbb{N}}$, respectively.

Proposition 1 ([14]): If $\mathbf{L} = \mathbf{L}_{\mathbb{C}}$, (10) *yields:*

$$y\left(\widetilde{\boldsymbol{x}}\right) = \arg\max_{i \in \mathbb{N}_c} \sum_{\boldsymbol{x}_j \in \widetilde{\mathcal{N}}} \mathcal{K}\left(\widetilde{\boldsymbol{x}}, \boldsymbol{x}_j\right)\mathbf{F}_{ji} \tag{11}$$

Proof: By construction and Assumption 1, we have:

$$\left(\widetilde{\mathbf{L}}_{\mathbb{C}}\right)_{\mathcal{U}\mathcal{U}} = \sum_{\boldsymbol{x}_k \in \widetilde{\mathcal{N}}} \mathcal{K}(\widetilde{\boldsymbol{x}}, \boldsymbol{x}_k)$$
$$\left(\left(\widetilde{\mathbf{L}}_{\mathbb{C}}\right)_{\mathcal{U}\mathcal{L}}\right)_j = -\mathcal{K}(\widetilde{\boldsymbol{x}}, \boldsymbol{x}_j)\widetilde{\mathbf{N}}_j$$

Therefore, (10) yields:

$$\begin{aligned} y\left(\widetilde{\boldsymbol{x}}\right) &= \arg\max_{i \in \mathbb{N}_c} \left(-\left(\widetilde{\mathbf{L}}_{\mathbb{C}}\right)_{\mathcal{U}\mathcal{U}}^{-1}\left(\widetilde{\mathbf{L}}_{\mathbb{C}}\right)_{\mathcal{U}\mathcal{L}}\mathbf{F}\right)_i \\ &= \arg\max_{i \in \mathbb{N}_c} \frac{\sum_{\boldsymbol{x}_j \in \widetilde{\mathcal{N}}} \mathcal{K}\left(\widetilde{\boldsymbol{x}}, \boldsymbol{x}_j\right)\mathbf{F}_{ji}}{\sum_{\boldsymbol{x}_k \in \widetilde{\mathcal{N}}} \mathcal{K}\left(\widetilde{\boldsymbol{x}}, \boldsymbol{x}_k\right)} \\ &= \arg\max_{i \in \mathbb{N}_c} \sum_{\boldsymbol{x}_j \in \widetilde{\mathcal{N}}} \mathcal{K}\left(\widetilde{\boldsymbol{x}}, \boldsymbol{x}_j\right)\mathbf{F}_{ji} \end{aligned}$$

■

Proposition 2: *If* $\mathbf{L} = \mathbf{L}_{\mathbb{N}}$, (10) *yields:*

$$y(\widetilde{\boldsymbol{x}}) = \arg\max_{i \in \mathbb{N}_c} \sum_{\boldsymbol{x}_j \in \widetilde{\mathcal{N}}} \frac{\mathcal{K}(\widetilde{\boldsymbol{x}}, \boldsymbol{x}_j)\,\mathbf{F}_{ji}}{\sqrt{\mathbf{D}_{jj} + \mathcal{K}(\widetilde{\boldsymbol{x}}, \boldsymbol{x}_j)}} \quad (12)$$

Proof: By construction, $\left(\widetilde{\mathbf{L}}_{\mathbb{N}}\right)_{\mathcal{U}\mathcal{U}} = 1$. From Assumption 1, we have:

$$\begin{aligned}\left(\left(\widetilde{\mathbf{L}}_{\mathbb{N}}\right)_{\mathcal{U}\mathcal{L}}\right)_j &= -\frac{\left(\widetilde{\mathbf{W}}_{\mathcal{U}\mathcal{L}}\right)_j}{\sqrt{\widetilde{\mathbf{D}}_{\mathcal{U}\mathcal{U}}}\sqrt{\left(\widetilde{\mathbf{D}}_{\mathcal{L}\mathcal{L}}\right)_{jj}}} \\ &= -\frac{\mathcal{K}(\widetilde{\boldsymbol{x}}, \boldsymbol{x}_j)\widetilde{\mathbf{N}}_j}{\sqrt{\sum_{\boldsymbol{x}_k \in \widetilde{\mathcal{N}}} \mathcal{K}(\widetilde{\boldsymbol{x}}, \boldsymbol{x}_k)}\sqrt{\mathbf{D}_{jj} + \mathcal{K}(\widetilde{\boldsymbol{x}}, \boldsymbol{x}_j)\widetilde{\mathbf{N}}_j}}\end{aligned} \quad (13)$$

Therefore, (10) yields:

$$\begin{aligned} y(\widetilde{\boldsymbol{x}}) &= \arg\max_{i \in \mathbb{N}_c} \left(-\left(\widetilde{\mathbf{L}}_{\mathbb{N}}\right)^{-1}_{\mathcal{U}\mathcal{U}} \left(\widetilde{\mathbf{L}}_{\mathbb{N}}\right)_{\mathcal{U}\mathcal{L}} \mathbf{F} \right)_i \\ &= \arg\max_{i \in \mathbb{N}_c} -\sum_{\boldsymbol{x}_j \in \widetilde{\mathcal{N}}} \left(\left(\widetilde{\mathbf{L}}_{\mathbb{N}}\right)_{\mathcal{U}\mathcal{L}}\right)_j \mathbf{F}_{ji} \\ &= \arg\max_{i \in \mathbb{N}_c} \sum_{\boldsymbol{x}_j \in \widetilde{\mathcal{N}}} \frac{\mathcal{K}(\widetilde{\boldsymbol{x}}, \boldsymbol{x}_j)\,\mathbf{F}_{ji}}{\sqrt{\mathbf{D}_{jj} + \mathcal{K}(\widetilde{\boldsymbol{x}}, \boldsymbol{x}_j)}} \end{aligned} \quad (14)$$

■

Proposition 1 shows that (10) is equivalent to the *Nadaraya-Watson kernel regression*[6] [23] for multi-class problems when $\mathbf{L} = \mathbf{L}_{\mathbb{C}}$. Proposition 2 provides a similar classification function than in Proposition 1, incorporating the additional normalization factor $\sqrt{\mathbf{D}_{jj} + \mathcal{K}(\widetilde{\boldsymbol{x}}, \boldsymbol{x}_j)}$. Therefore, we classify $\widetilde{\boldsymbol{x}}$ as:

$$y(\widetilde{\boldsymbol{x}}) = \arg\max_{i \in \mathbb{N}_c} \sum_{\boldsymbol{x}_j \in \widetilde{\mathcal{N}}} \mathbf{\Gamma}_{\mathbf{L}}(\widetilde{\boldsymbol{x}}, \boldsymbol{x}_j)\,\mathbf{F}_{ji}$$

such that

$$\mathbf{\Gamma}_{\mathbf{L}}(\widetilde{\boldsymbol{x}}, \boldsymbol{x}_j) = \begin{cases} \mathcal{K}(\widetilde{\boldsymbol{x}}, \boldsymbol{x}_j), & \mathbf{L} = \mathbf{L}_{\mathbb{C}} \\ \mathcal{K}(\widetilde{\boldsymbol{x}}, \boldsymbol{x}_j)\left(\mathbf{D}_{jj} + \mathcal{K}(\widetilde{\boldsymbol{x}}, \boldsymbol{x}_j)\right)^{-1/2}, & \mathbf{L} = \mathbf{L}_{\mathbb{N}} \end{cases}$$

IV. Proposed kernelized methods

This section formulates the K-GFHF and K-CGFHF methods. Assume that $\left(\mathbf{A}^{-1}\right)_{\bullet\circ}$ is the submatrix $\bullet\circ$ of $\mathbf{A}^{-1}$ while $\mathbf{A}^{-1}_{\bullet\bullet}$ is the inverse of $\mathbf{A}_{\bullet\bullet}$, such that $\bullet$ and $\circ$ can be even $\mathcal{L}$ or $\mathcal{U}$. We start our theoretical analysis with Proposition 3, which is a well known block matrix inversion property.

Proposition 3: *If* $\mathbf{K}$ *is symmetric and invertible, we have:*

$$\begin{aligned} \left(\mathbf{K}^{-1}\right)_{\mathcal{L}\mathcal{L}} &= \left(\mathbf{K}_{\mathcal{L}\mathcal{L}} - \mathbf{K}_{\mathcal{U}\mathcal{L}}^\top \mathbf{K}_{\mathcal{U}\mathcal{U}}^{-1} \mathbf{K}_{\mathcal{U}\mathcal{L}}\right)^{-1} \\ \left(\mathbf{K}^{-1}\right)_{\mathcal{L}\mathcal{U}} &= -\left(\mathbf{K}_{\mathcal{L}\mathcal{L}} - \mathbf{K}_{\mathcal{U}\mathcal{L}}^\top \mathbf{K}_{\mathcal{U}\mathcal{U}}^{-1} \mathbf{K}_{\mathcal{U}\mathcal{L}}\right)^{-1} \mathbf{K}_{\mathcal{U}\mathcal{L}}^\top \mathbf{K}_{\mathcal{U}\mathcal{U}}^{-1} \\ \left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{L}} &= -\left(\mathbf{K}_{\mathcal{U}\mathcal{U}} - \mathbf{K}_{\mathcal{U}\mathcal{L}} \mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1} \mathbf{K}_{\mathcal{U}\mathcal{L}}^\top\right)^{-1} \mathbf{K}_{\mathcal{U}\mathcal{L}} \mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1} \\ \left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}} &= \left(\mathbf{K}_{\mathcal{U}\mathcal{U}} - \mathbf{K}_{\mathcal{U}\mathcal{L}} \mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1} \mathbf{K}_{\mathcal{U}\mathcal{L}}^\top\right)^{-1} \end{aligned} \quad (15)$$

Proof: Since $\mathbf{K} = \mathbf{K}^\top$ and $\mathbf{K}\mathbf{K}^{-1} = \mathbf{I}_n$, we have:

$$\begin{aligned} &\begin{bmatrix} \mathbf{K}_{\mathcal{L}\mathcal{L}} & \mathbf{K}_{\mathcal{U}\mathcal{L}}^\top \\ \mathbf{K}_{\mathcal{U}\mathcal{L}} & \mathbf{K}_{\mathcal{U}\mathcal{U}} \end{bmatrix} \begin{bmatrix} \left(\mathbf{K}^{-1}\right)_{\mathcal{L}\mathcal{L}} & \left(\mathbf{K}^{-1}\right)_{\mathcal{L}\mathcal{U}} \\ \left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{L}} & \left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}} \end{bmatrix} \\ &\quad = \begin{bmatrix} \mathbf{I}_l & \mathbf{O}_{l \times u} \\ \mathbf{O}_{u \times l} & \mathbf{I}_u \end{bmatrix} \end{aligned}$$

With simple algebra, we get (15).

■

From the kernel expansions in [22], we have $\mathbf{F} = \mathbf{K}\boldsymbol{\alpha}$ with $\boldsymbol{\alpha} := \begin{bmatrix} \boldsymbol{\alpha}_{\mathcal{L}} \\ \boldsymbol{\alpha}_{\mathcal{U}} \end{bmatrix} \in \mathbb{R}^{n \times c}$. By definition, $\boldsymbol{\alpha}_{\mathcal{L}} \in \mathbb{R}^l$ and $\boldsymbol{\alpha}_{\mathcal{U}} \in \mathbb{R}^u$. Assuming that $\mathbf{F}_{\mathcal{L}} = \mathbf{Y}_{\mathcal{L}}$, we have:

$$\begin{aligned} \begin{bmatrix} \mathbf{F}_{\mathcal{L}} \\ \mathbf{F}_{\mathcal{U}} \end{bmatrix} &= \begin{bmatrix} \mathbf{K}_{\mathcal{L}\mathcal{L}} & \mathbf{K}_{\mathcal{U}\mathcal{L}}^\top \\ \mathbf{K}_{\mathcal{U}\mathcal{L}} & \mathbf{K}_{\mathcal{U}\mathcal{U}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_{\mathcal{L}} \\ \boldsymbol{\alpha}_{\mathcal{U}} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{K}_{\mathcal{L}\mathcal{L}}\boldsymbol{\alpha}_{\mathcal{L}} + \mathbf{K}_{\mathcal{U}\mathcal{L}}^\top \boldsymbol{\alpha}_{\mathcal{U}} \\ \mathbf{K}_{\mathcal{U}\mathcal{L}}\boldsymbol{\alpha}_{\mathcal{L}} + \mathbf{K}_{\mathcal{U}\mathcal{U}}\boldsymbol{\alpha}_{\mathcal{U}} \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_{\mathcal{L}} \\ \mathbf{F}_{\mathcal{U}} \end{bmatrix} \end{aligned} \quad (16)$$

Therefore,

$$\boldsymbol{\alpha}_{\mathcal{L}} = \mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1}\left(\mathbf{Y}_{\mathcal{L}} - \mathbf{K}_{\mathcal{U}\mathcal{L}}^\top \boldsymbol{\alpha}_{\mathcal{U}}\right) \quad (17)$$

From (16) and (17), we obtain:

$$\begin{aligned} \mathbf{F}_{\mathcal{U}} &= \mathbf{K}_{\mathcal{U}\mathcal{L}}\boldsymbol{\alpha}_{\mathcal{L}} + \mathbf{K}_{\mathcal{U}\mathcal{U}}\boldsymbol{\alpha}_{\mathcal{U}} \\ &= \mathbf{K}_{\mathcal{U}\mathcal{L}}\mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1}\mathbf{Y}_{\mathcal{L}} + \left(\mathbf{K}^{-1}\right)^{-1}_{\mathcal{U}\mathcal{U}} \boldsymbol{\alpha}_{\mathcal{U}} \end{aligned} \quad (18)$$

A. K-GFHF

The optimization problem in (1) can be rewritten as[7]:

$$\min_{\mathbf{F}_{\mathcal{U}} \in \mathbb{R}^{u \times c}} \operatorname{tr}\left(\mathbf{F}_{\mathcal{U}}^\top \mathbf{L}_{\mathcal{U}\mathcal{U}} \mathbf{F}_{\mathcal{U}}\right) + 2\operatorname{tr}\left(\mathbf{F}_{\mathcal{U}}^\top \mathbf{L}_{\mathcal{U}\mathcal{L}} \mathbf{Y}_{\mathcal{L}}\right) \quad (19)$$

From (18), we have:

$$\begin{aligned} \operatorname{tr}\left(\mathbf{F}_{\mathcal{U}}^\top \mathbf{L}_{\mathcal{U}\mathcal{L}} \mathbf{Y}_{\mathcal{L}}\right) &= \operatorname{tr}\left(\boldsymbol{\alpha}_{\mathcal{U}}^\top \left(\mathbf{K}^{-1}\right)^{-1}_{\mathcal{U}\mathcal{U}} \mathbf{L}_{\mathcal{U}\mathcal{L}} \mathbf{Y}_{\mathcal{L}}\right) \\ &\quad + \operatorname{tr}\left(\mathbf{Y}_{\mathcal{L}}^\top \mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1} \mathbf{K}_{\mathcal{U}\mathcal{L}}^\top \mathbf{L}_{\mathcal{U}\mathcal{L}} \mathbf{Y}_{\mathcal{L}}\right) \end{aligned}$$

[6]Given a kernel function $K_h(\cdot)$ with bandwidth $h > 0$, a sample $\{\boldsymbol{x}_i\}_{i=1}^n \subset \mathbb{R}^d$, and target values $\{y_i\}_{i=1}^n \subset \mathbb{R}$, the Nadaraya-Watson kernel regression provides an estimator $\widehat{m}_h(\boldsymbol{x}) = \left[\sum_{i=1}^n K_h(\boldsymbol{x} - \boldsymbol{x}_i) y_i\right] / \sum_{i=1}^n K_h(\boldsymbol{x} - \boldsymbol{x}_i)$.

[7]See [14] for details.

and

$$\mathrm{tr}\left(\mathbf{F}_{\mathcal{U}}^{\top}\mathbf{L}_{\mathcal{U}\mathcal{U}}\mathbf{F}_{\mathcal{U}}\right) = \mathrm{tr}\left(\mathbf{Y}_{\mathcal{L}}^{\top}\mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1}\mathbf{K}_{\mathcal{U}\mathcal{L}}^{\top}\mathbf{L}_{\mathcal{U}\mathcal{U}}\mathbf{K}_{\mathcal{U}\mathcal{L}}\mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1}\mathbf{Y}_{\mathcal{L}}\right) + \mathrm{tr}\left(\boldsymbol{\alpha}_{\mathcal{U}}^{\top}\left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{U}}\left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}^{-1}\boldsymbol{\alpha}_{\mathcal{U}}\right) + 2\mathrm{tr}\left(\boldsymbol{\alpha}_{\mathcal{U}}^{\top}\left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{U}}\mathbf{K}_{\mathcal{U}\mathcal{L}}\mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1}\mathbf{Y}_{\mathcal{L}}\right)$$

Therefore, the optimization problem in (19) becomes:

$$\min_{\boldsymbol{\alpha}_{\mathcal{U}}\in\mathbb{R}^{u\times c}} \boldsymbol{\Gamma}(\boldsymbol{\alpha}_{\mathcal{U}}) \tag{20}$$

such that

$$\boldsymbol{\Gamma}(\boldsymbol{\alpha}_{\mathcal{U}}) = \mathrm{tr}\left(\boldsymbol{\alpha}_{\mathcal{U}}^{\top}\left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{U}}\left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}^{-1}\boldsymbol{\alpha}_{\mathcal{U}}\right) + 2\mathrm{tr}\left(\boldsymbol{\alpha}_{\mathcal{U}}^{\top}\left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}}\right) + 2\mathrm{tr}\left(\boldsymbol{\alpha}_{\mathcal{U}}^{\top}\left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{U}}\mathbf{K}_{\mathcal{U}\mathcal{L}}\mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1}\mathbf{Y}_{\mathcal{L}}\right)$$

The closed-form solution of (20) is given by:

$$\begin{aligned}\boldsymbol{\alpha}_{\mathcal{U}} &= -\left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}\left(\mathbf{K}_{\mathcal{U}\mathcal{L}}\mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1} + \mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\right)\mathbf{Y}_{\mathcal{L}} \\ &= \left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}} - \left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}}\end{aligned} \tag{21}$$

Substituting (21) in (17), we obtain:

$$\begin{aligned}\boldsymbol{\alpha}_{\mathcal{L}} &= \mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1}\mathbf{Y}_{\mathcal{L}} - \mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1}\mathbf{K}_{\mathcal{U}\mathcal{L}}^{\top}\left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}} \\ &\quad + \mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1}\mathbf{K}_{\mathcal{U}\mathcal{L}}^{\top}\left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}} \\ &= \left[\mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1} + \mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1}\mathbf{K}_{\mathcal{U}\mathcal{L}}^{\top}\mathbf{K}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{K}_{\mathcal{U}\mathcal{L}}\left(\mathbf{K}^{-1}\right)_{\mathcal{L}\mathcal{L}}\right]\mathbf{Y}_{\mathcal{L}} \\ &\quad - \left(\mathbf{K}^{-1}\right)_{\mathcal{L}\mathcal{U}}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}}\end{aligned} \tag{22}$$

From the first equation in (15), we have:

$$\mathbf{K}_{\mathcal{U}\mathcal{L}}^{\top}\mathbf{K}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{K}_{\mathcal{U}\mathcal{L}} = \mathbf{K}_{\mathcal{L}\mathcal{L}} - \left(\mathbf{K}^{-1}\right)_{\mathcal{L}\mathcal{L}}^{-1}$$

Therefore, (22) yields:

$$\boldsymbol{\alpha}_{\mathcal{L}} = \left(\mathbf{K}^{-1}\right)_{\mathcal{L}\mathcal{L}}\mathbf{Y}_{\mathcal{L}} - \left(\mathbf{K}^{-1}\right)_{\mathcal{L}\mathcal{U}}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}} \tag{23}$$

From (21) and (23), we have:

$$\begin{aligned}\boldsymbol{\alpha} &= \begin{bmatrix} \left(\mathbf{K}^{-1}\right)_{\mathcal{L}\mathcal{L}}\mathbf{Y}_{\mathcal{L}} - \left(\mathbf{K}^{-1}\right)_{\mathcal{L}\mathcal{U}}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}} \\ \left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}} - \left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}} \end{bmatrix} \\ &= \begin{bmatrix} \left(\mathbf{K}^{-1}\right)_{\mathcal{L}\mathcal{L}} & \left(\mathbf{K}^{-1}\right)_{\mathcal{L}\mathcal{U}} \\ \left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{L}} & \left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}} \end{bmatrix}\begin{bmatrix} \mathbf{Y}_{\mathcal{L}} \\ -\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}} \end{bmatrix} \\ &= \mathbf{K}^{-1}\begin{bmatrix} \mathbf{Y}_{\mathcal{L}} \\ -\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}} \end{bmatrix}\end{aligned} \tag{24}$$

Eq. (24) shows that K-GFHF and GFHF have the same transductive solution, independently of $\mathbf{K}$. However, K-GFHF has the advantage of being an inductive method.

B. K-CGFHF

K-CGFHF has the same objective function in (20), incorporating the normalization constraints in [17]. The normalization constraints $\mathbf{F}\mathbf{1}_c = \mathbf{1}_n$ and $\mathbf{F}^{\top}\mathbf{1}_n = n\boldsymbol{\omega}$, subject to $\mathbf{F}_{\mathcal{L}} = \mathbf{Y}_{\mathcal{L}}$, are equivalent to:

$$\mathbf{F}_{\mathcal{U}}\mathbf{1}_c = \mathbf{1}_u, \quad \mathbf{F}_{\mathcal{U}}^{\top}\mathbf{1}_u = n\boldsymbol{\omega} - \mathbf{Y}_{\mathcal{L}}^{\top}\mathbf{1}_l \tag{25}$$

From (18) and (25), we get the following constraints:

$$\begin{aligned}\boldsymbol{\alpha}_{\mathcal{U}}\mathbf{1}_c &= \left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}\left(\mathbf{1}_u - \mathbf{K}_{\mathcal{U}\mathcal{L}}\mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1}\mathbf{1}_l\right) \\ \boldsymbol{\alpha}_{\mathcal{U}}^{\top}\left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u &= n\boldsymbol{\omega} - \mathbf{Y}_{\mathcal{L}}^{\top}\mathbf{1}_l - \mathbf{Y}_{\mathcal{L}}^{\top}\mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1}\mathbf{K}_{\mathcal{U}\mathcal{L}}^{\top}\mathbf{1}_u\end{aligned} \tag{26}$$

From (20) and (26), we can formulate K-CGFHF as the following constrained optimization problem:

$$\begin{aligned}&\min_{\boldsymbol{\alpha}_{\mathcal{U}}\in\mathbb{R}^{u\times c}} \boldsymbol{\Gamma}(\boldsymbol{\alpha}_{\mathcal{U}}) \\ &\text{s.t.}\begin{cases} \boldsymbol{\alpha}_{\mathcal{U}}\mathbf{1}_c = \left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}\left(\mathbf{1}_u - \mathbf{K}_{\mathcal{U}\mathcal{L}}\mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1}\mathbf{1}_l\right) \\ \boldsymbol{\alpha}_{\mathcal{U}}^{\top}\left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u = n\boldsymbol{\omega} - \mathbf{Y}_{\mathcal{L}}^{\top}\mathbf{1}_l - \mathbf{Y}_{\mathcal{L}}^{\top}\mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1}\mathbf{K}_{\mathcal{U}\mathcal{L}}^{\top}\mathbf{1}_u \end{cases}\end{aligned} \tag{27}$$

Proposition 4 provides the closed-form solution of (27) while Corollary 1 proves that K-CGFHF generalizes RMGT and RMGTHOR.

Proposition 4: The closed-form solution of (27) *is given by:*

$$\begin{aligned}\boldsymbol{\alpha}_{\mathcal{U}} &= \left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}} \\ &\quad + \left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}\left[-\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}} + \frac{\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u}{\mathbf{1}_u^{\top}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u}\boldsymbol{\tau} + \frac{1}{c}\boldsymbol{\nu}\mathbf{1}_c^{\top}\right]\end{aligned} \tag{28}$$

where:

$$\begin{aligned}\boldsymbol{\tau} &= n\boldsymbol{\omega}^{\top} - \mathbf{1}_l^{\top}\mathbf{Y}_{\mathcal{L}} + \mathbf{1}_u^{\top}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}} \\ \boldsymbol{\nu} &= \mathbf{1}_u + \mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{1}_l - \frac{\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u}{\mathbf{1}_u^{\top}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u}\left(u + \mathbf{1}_u^{\top}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{1}_l\right)\end{aligned}$$

Proof: The Lagrangian corresponding to (27) is given by:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\alpha}_{\mathcal{U}}, \boldsymbol{\xi}, \boldsymbol{\lambda}) &= \boldsymbol{\Gamma}(\boldsymbol{\alpha}_{\mathcal{U}}) \\ &\quad - \boldsymbol{\xi}^{\top}\left(\boldsymbol{\alpha}_{\mathcal{U}}\mathbf{1}_c - \left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}\left(\mathbf{1}_u - \mathbf{K}_{\mathcal{U}\mathcal{L}}\mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1}\mathbf{1}_l\right)\right) \\ &\quad - \boldsymbol{\lambda}^{\top}\left(\boldsymbol{\alpha}_{\mathcal{U}}^{\top}\left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u - n\boldsymbol{\omega} + \mathbf{Y}_{\mathcal{L}}^{\top}\mathbf{1}_l \right. \\ &\quad \left. + \mathbf{Y}_{\mathcal{L}}^{\top}\mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1}\mathbf{K}_{\mathcal{U}\mathcal{L}}^{\top}\mathbf{1}_u\right)\end{aligned}$$

where $\boldsymbol{\xi} \in \mathbb{R}^u$ and $\boldsymbol{\lambda} \in \mathbb{R}^c$ are the Lagrange multipliers. Zeroing $\partial\mathcal{L}/\partial\boldsymbol{\alpha}_{\mathcal{U}}$, we obtain:

$$\boldsymbol{\alpha}_{\mathcal{U}} = \left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}} \mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1} \left[- \left(\mathbf{L}_{\mathcal{U}\mathcal{U}}\mathbf{K}_{\mathcal{U}\mathcal{L}}\mathbf{K}_{\mathcal{L}\mathcal{L}}^{-1} + \mathbf{L}_{\mathcal{U}\mathcal{L}}\right)\mathbf{Y}_{\mathcal{L}} + \frac{1}{2}\left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}} \boldsymbol{\xi}\mathbf{1}_c^{\top} + \frac{1}{2}\mathbf{1}_u\boldsymbol{\lambda}^{\top}\right] \tag{29}$$

Substituting (29) in the first constraint, we obtain:

$$\frac{c}{2}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}\boldsymbol{\xi} = \mathbf{1}_u + \mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{1}_l - \frac{1}{2}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u\boldsymbol{\lambda}^{\top}\mathbf{1}_c \tag{30}$$

Substituting (29) in the second constraint, we obtain:

$$\boldsymbol{\lambda}^{\top} = \frac{2}{\mathbf{1}_u^{\top}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u}\left[n\boldsymbol{\omega}^{\top} - \mathbf{1}_l^{\top}\mathbf{Y}_{\mathcal{L}} + \mathbf{1}_{\mathcal{U}}^{\top}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}} - \frac{1}{2}\mathbf{1}_u^{\top}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}\boldsymbol{\xi}\mathbf{1}_c\right] \tag{31}$$

Substituting (31) in (30), we obtain:

$$\boldsymbol{\xi} = \frac{2}{c}\boldsymbol{\varpi}^{-1}\left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}\left[\mathbf{1}_u + \mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{1}_l - \frac{\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u}{\mathbf{1}_u^{\top}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u}\left(u + \mathbf{1}_u^{\top}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{1}_l\right)\right] \tag{32}$$

in which

$$\boldsymbol{\varpi} = \left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}} - \frac{\left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u\mathbf{1}_u^{\top}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}}{\mathbf{1}_u^{\top}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u}$$

Substituting (31) in (29), we obtain:

$$\boldsymbol{\alpha}_{\mathcal{U}} = -\left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}} - \left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}} + \frac{1}{2}\boldsymbol{\varpi}\boldsymbol{\xi}\mathbf{1}_c^{\top} + \frac{\left(\mathbf{K}^{-1}\right)_{\mathcal{U}\mathcal{U}}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u}{\mathbf{1}_u^{\top}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u}\times \times\left(n\boldsymbol{\omega}^{\top} - \mathbf{1}_l^{\top}\mathbf{Y}_{\mathcal{L}} + \mathbf{1}_u^{\top}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}}\right) \tag{33}$$

Substituting (32) in (33), we get (28).

■

Corollary 1: From (28) *and* (17)*, we have:*

$$\boldsymbol{\alpha} = \mathbf{K}^{-1}\begin{bmatrix} \mathbf{Y}_{\mathcal{L}} \\ -\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}} + \frac{\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u}{\mathbf{1}_u^{\top}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u}\boldsymbol{\tau} + \frac{1}{c}\boldsymbol{\nu}\mathbf{1}_c^{\top} \end{bmatrix} \tag{34}$$

in which

$$\boldsymbol{\tau} = n\boldsymbol{\omega}^{\top} - \mathbf{1}_l^{\top}\mathbf{Y}_{\mathcal{L}} + \mathbf{1}_u^{\top}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}}$$

$$\boldsymbol{\nu} = \mathbf{1}_u + \mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{1}_l - \frac{\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u}{\mathbf{1}_u^{\top}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u}\left(u + \mathbf{1}_u^{\top}\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{1}_l\right)$$

Moreover, if $\mathbf{L} = \mathbf{L}_{\mathbb{C}}$, (34) *yields:*

$$\boldsymbol{\alpha} = \mathbf{K}^{-1}\begin{bmatrix} \mathbf{Y}_{\mathcal{L}} \\ -(\mathbf{L}_{\mathbb{C}})_{\mathcal{U}\mathcal{U}}^{-1}(\mathbf{L}_{\mathbb{C}})_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}} + \frac{(\mathbf{L}_{\mathbb{C}})_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u}{\mathbf{1}_u^{\top}(\mathbf{L}_{\mathbb{C}})_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{1}_u}\boldsymbol{\tau} \end{bmatrix} \tag{35}$$

in which

$$\boldsymbol{\tau} = n\boldsymbol{\omega}^{\top} - \mathbf{1}_l^{\top}\mathbf{Y}_{\mathcal{L}} + \mathbf{1}_u^{\top}(\mathbf{L}_{\mathbb{C}})_{\mathcal{U}\mathcal{U}}^{-1}(\mathbf{L}_{\mathbb{C}})_{\mathcal{U}\mathcal{L}}\mathbf{Y}_{\mathcal{L}}$$

Proof: (34) comes directly from (28) and (17). From (34) and Corollary 1 in [18], we get (35).

■

V. EXPERIMENTAL EVALUATION

We performed an experimental evaluation to verify the effectiveness of K-GFHF and K-CGFHF on inductive SSL tasks on benchmark data sets. We compared our methods against LapRLS, LapSVM, and H-GFHF (using both $\mathbf{L}_{\mathbb{C}}$ and $\mathbf{L}_{\mathbb{N}}$). We used a variation of the code in [20][8] for these methods. Table I describes the data sets used in our experiments, which are freely available[9] and widely used in the SSL literature [4].

TABLE I. DESCRIPTION OF THE DATA SETS.

Data sets	n	d	c	% minority class	% majority class
COIL$_2$	1500	241	2	50.0	50.0
DIGIT-1	1500	241	2	48.93	51.07
G-241C	1500	241	2	50.0	50.0
G-241N	1500	241	2	49.87	50.13
USPS	1500	241	2	20.0	80.0

A. Setup

From the training sample $\mathcal{X}$, we generated a distance matrix $\boldsymbol{\Psi} \in \mathbb{R}^{n \times n}$ using the l_2-norm (or Euclidean distance) as distance function for all data sets. From $\boldsymbol{\Psi}$, we generated an adjacency matrix $\mathbf{A} \in \mathbb{B}^{n \times n}$ using the *symmetric k-nearest neighbors* (symKNN) graph, which is widely used for graph construction [20]. In the symKNN graph, there exists an undirected edge between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ if $\boldsymbol{x}_j$ is one of the k closest examples of $\boldsymbol{x}_i$ or vice versa. We set $k = 10$.

From $\mathbf{A}$ and $\boldsymbol{\Psi}$, we generated a weighted matrix $\mathbf{W}$ using the *radial basis function* (RBF) kernel, which is defined by:

$$\mathcal{K}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \exp\left(-\frac{\Psi^2\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)}{2\sigma^2}\right) \tag{36}$$

where $\sigma \in \mathbb{R}_+^*$ is the kernel bandwidth parameter. Considering $\boldsymbol{x}_i^{(k)} \in \mathcal{X}$ the k-th nearest neighbor of $\boldsymbol{x}_i$ in $\mathcal{X}$, we estimated the value of σ by:

$$\sigma = \frac{1}{3n}\sum_{i=1}^{n}\Psi\left(\boldsymbol{x}_i, \boldsymbol{x}_i^{(k)}\right) \tag{37}$$

[8] http://sites.labic.icmc.usp.br/sousa/experiments_graph_SSL/.
[9] http://olivier.chapelle.cc/ssl-book/benchmarks.html.

TABLE II. AVERAGE ERROR RATES (%) AND STANDARD DEVIATIONS (%) FOR THE SSL ALGORITHMS ON INDUCTIVE SSL TASKS.

	in-sample error					
	$COIL_2$	DIGIT-1	G-241C	G-241N	USPS	ranking
LapRLS	41.27 (6.75)	10.38 (6.36)	49.50 (1.54)	48.95 (2.17)	16.66 (7.63)	3.6
LapSVM	40.97 (5.61)	14.90 (5.64)	49.54 (1.49)	49.02 (2.00)	18.88 (7.14)	4.6
H-GFHF($\mathbf{L}_{\mathbb{C}}$)	46.42 (6.58)	15.49 (9.85)	49.64 (1.28)	49.45 (1.30)	16.97 (4.09)	5.8
H-GFHF($\mathbf{L}_{\mathbb{N}}$)	46.00 (6.75)	10.29 (6.43)	49.44 (1.61)	48.87 (2.21)	**14.24 (5.19)**	2.6
K-GFHF	46.00 (6.75)	10.29 (6.43)	49.44 (1.61)	48.87 (2.21)	16.77 (7.76)	3.2
K-CGFHF	**34.30 (4.94)**	**7.35 (3.79)**	**42.58 (3.29)**	**46.81 (3.65)**	16.21 (6.88)	**1.2**
	out-of-sample error					
	$COIL_2$	DIGIT-1	G-241C	G-241N	USPS	ranking
LapRLS	38.85 (5.80)	10.91 (6.61)	49.80 (1.27)	49.50 (2.09)	17.37 (7.46)	4.8
LapSVM	38.82 (5.84)	10.92 (6.57)	49.74 (1.50)	49.42 (2.25)	17.45 (7.31)	4.4
H-GFHF($\mathbf{L}_{\mathbb{C}}$)	40.93 (6.73)	15.45 (9.79)	49.60 (1.14)	49.46 (1.29)	16.95 (4.00)	4.2
H-GFHF($\mathbf{L}_{\mathbb{N}}$)	40.89 (6.98)	10.25 (6.40)	49.61 (1.27)	49.25 (1.92)	**14.42 (4.97)**	2.6
K-GFHF	37.31 (5.98)	10.82 (6.66)	49.73 (1.54)	49.41 (2.25)	17.47 (7.57)	3.6
K-CGFHF	**34.32 (5.26)**	**7.50 (3.99)**	**35.64 (4.46)**	**47.18 (5.07)**	16.99 (6.51)	**1.4**

as suggested in [24].

From $\mathbf{W}$, we generated a graph Laplacian $\mathbf{L}$. Since the normalized Laplacian $\mathbf{L}_{\mathbb{N}}$ may lead to better results in comparison to the combinatorial Laplacian $\mathbf{L}_{\mathbb{C}}$ [19], we used $\mathbf{L}_{\mathbb{N}}$ as "basis" Laplacian when applicable. We generated $\mathbf{L}_{\mathbb{C}}$ and $\mathbf{L}_{\mathbb{N}}$ as $\mathbf{L}_{\mathbb{C}} = \eta\mathbf{D} - \mathbf{W}$ and $\mathbf{L}_{\mathbb{N}} = \eta\mathbf{I}_n - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$, respectively, as suggested in [20]. A small $\eta > 1$ is used to increase the eigenvalues of the graph Laplacians in an attempt to overcome numerical instabilities while solving linear systems using these Laplacians. We set $\eta = 1.01$, as suggested in [20].

Since *higher order regularization* is effective on SSL tasks [25], we used in LapRLS, LapSVM, K-GFHF, and K-CGFHF the *iterated* Laplacian $\mathbf{L}_{\mathbb{I}} = \mathbf{L}_{\mathbb{N}}^p$ with $p \in \mathbb{N}^*$. We set $p = 2$, as suggested in [18].

The kernel matrix $\mathbf{K}$ was generated from $\mathcal{X}$ using the RBF kernel, as in (36). We used the same parameter estimation in (37) to choose the value of σ.

We ran LapSVM using the source code in [26][10] with the default values for code parameters. We trained LapSVM in the primal using Newton's method instead of preconditioned conjugate gradient, as suggested in [20]. For LapRLS and LapSVM, we set $\gamma_A = 10^{-6}$, $\gamma_I = 10^{-2}$, which achieved the best results during preliminary experiments.

Assume that $\widetilde{k} \in \mathbb{N}^*$ is the number of neighbors of $\widetilde{\boldsymbol{x}}$ in $\mathcal{X}$, which should be small and usually $\widetilde{k} \leq k$. For H-GFHF, we set $\widetilde{k} = 5$ for both $\mathbf{L}_{\mathbb{C}}$ and $\mathbf{L}_{\mathbb{N}}$. For K-CGFHF, we set $\boldsymbol{\omega}$ as the class prior probabilities for USPS and the uniform class distribution ($\boldsymbol{\omega} = \mathbf{1}_c/c$) for the other data sets, as suggested in [20].

B. Evaluation

For a fair comparison, we used the same experimental protocol in [27]. Specifically, we performed 10 times 4-fold cross validation using each fold as a test set once. For each train/test split, 10 random choices of $l = 5c$ labeled examples were done in the current training set (three folds) without stratification, ensuring that there is at least one labeled example of each class.

[10]http://www.dii.unisi.it/~melacci/lapsvmp/index.html.

The SSL methods were trained in the current training set (containing labeled and unlabeled sets) and we evaluate their classification performance on the unlabeled and test sets. We call *in-sample error* and *out-of-sample error* the average error rates achieved by the methods on the unlabeled and test sets, respectively.

C. Results

Table II shows the classification performances of the SSL methods for both transduction and induction using the data sets described in Table I and the average rankings of the corresponding methods. The best result in each data set for both transduction and induction is marked in bold face. The three worst results in each data set for both transduction and induction have a grey background.

Since we were analyzing the relative rankings of many algorithms over many data sets, we ran the *Friedman's test*[11] with the *Bonferroni's post test* using a significance level of 0.05. We used the software *Orange*[12] to perform our statistical analysis. We set K-CGFHF as "*control*" method. The rankings referred to the methods that were statistically outperformed by our method are marked with a grey background.

From Table II, we see that K-CGFHF achieved the best result in 4 out of 5 data sets for both transduction and induction, being outperformed by H-GFHF($\mathbf{L}_{\mathbb{N}}$) in the USPS data set. On $COIL_2$, DIGIT-1, and G-241C, our results outperformed the competing methods by a large margin. Therefore, these results evidence the effectiveness of higher order regularization combined with the normalization constraints in [17] for both transductive and inductive SSL tasks using the RBF kernel.

Considering that a data set has *high unbalanced ratio* if the majority class has at least three times more examples than the minority class, the authors in [5] showed that RMGT may be uneffective on data sets with high unbalanced ratio. The authors provided experiments on a variety of unbalanced data sets from the time series domain. Detailed results can be found in [1].

[11]See [28] for a review on statistical tests for machine learning.

[12]http://orange.biolab.si/.

Since our method is a generalization of RMGT, such an issue may also occur on our method, independently of the kernel function used. This may explain the poor classification performance of our method on the USPS data set. Unfortunately, even with a careful parameter selection our method provided no satisfactory results on data sets with high unbalanced ratio.

LapSVM and H-GFHF($\mathbf{L}_{\mathbb{C}}$) were statistically outperformed by K-CGFHF on the unlabeled sets. However, only LapRLS was statistically outperformed by K-CGFHF on the test sets. Although our method frequently achieved better results than the competing methods, the statistical tests found significant differences only in a few cases. This is likely to be due to the use of a non-parametric statistical test which requires results in more data sets in order to detect additional significant differences.

VI. Conclusion

In this paper, we generalized GFHF, RMGT, and RMGTHOR for inductive SSL through the kernel expansions in [22]. The generalized methods (K-GFHF and K-CGFHF) yield the same transductive solution of the corresponding base methods, independently of the kernel function used. Therefore, we maintain the classification performance of the base methods on the unlabeled sets and provide inductive classification through kernel expansions over both labeled and unlabeled examples.

Moreover, K-GFHF and K-CGFHF are parameter-free. This is a great advantage over other SSL methods like LapRLS and LapSVM, which may require a careful parameter selection and may be strongly dependent of parameter selection on some data sets [29].

Through experiments on inductive SSL tasks on benchmark data sets, we showed the effectiveness of the proposed method for both transduction and induction. Specifically, we showed that our method achieved the best result on most data sets. Moreover, our method achieved the best average ranking for both in-sample and out-of-sample error, outperforming the competing methods.

However, our method may be uneffective on data sets with high unbalanced ratio, as its base method RMGT. A way to overcome this issue will be investigated in future research.

References

[1] C. A. R. Sousa, V. M. A. Souza, and G. E. A. P. A. Batista, "Time series transductive classification on graphs: supplementary material," 2020. [Online]. Available: https://www.researchgate.net/publication/316649218_Time_series_transductive_classification_on_graphs_supplementary_material

[2] C. A. R. Sousa and G. E. A. P. A. Batista, "Constrained graph-based semi-supervised learning: supplementary material," 2017. [Online]. Available: https://www.researchgate.net/publication/316634288_Constrained_graph-based_semi-supervised_learning_supplementary_material

[3] X. Zhu, "Semi-supervised learning literature survey," Computer Sciences, University of Wisconsin-Madison, Tech. Rep. 1530, 2005.

[4] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-supervised learning*. The MIT Press, 2006.

[5] C. A. R. Sousa, V. M. A. Souza, and G. E. A. P. A. Batista, "Time series transductive classification on imbalanced data sets: an experimental study," in *ICPR*, 2014, pp. 3780–3785.

[6] M. Belkin and P. Niyogi, "Towards a theoretical foundation for laplacian based manifold methods," *Journal of Computer and System Sciences*, vol. 74, no. 8, pp. 1289–1308, 2008.

[7] C. Cortes, M. Mohri, D. Pechyony, and A. Rastogi, "Stability of transductive regression algorithms," in *ICML*, 2008, pp. 176–183.

[8] T. Zhang and R. Ando, "Analysis of spectral kernel design based semi-supervised learning," in *NIPS*, 2006, pp. 1601–1608.

[9] C. A. R. Sousa, "Constrained graph-based semi-supervised learning with higher order regularization," Ph.D. dissertation, University of São Paulo, Brazil, 2017.

[10] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, 2019.

[11] O. Delalleau, Y. Bengio, and N. Roux, "Efficient non-parametric function induction in semi-supervised learning," in *AISTATS*, 2005, pp. 96–103.

[12] R. Coifman and S. Lafon, "Geometric harmonics: a novel tool for multiscale out-of-sample extension of empirical functions," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 31–52, 2006.

[13] C. K. I. Williams and M. Seeger, "Using the nyström method to speed up kernel machines," in *NIPS*, 2001, pp. 682–688.

[14] C. A. R. Sousa, "An overview on the gaussian fields and harmonic functions method for semi-supervised learning," in *IJCNN*, 2015, pp. 1439–1446.

[15] ——, "An inductive semi-supervised learning approach for the local and global consistency algorithm," in *IJCNN*, 2016.

[16] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, 2003, pp. 912–919.

[17] W. Liu and S.-F. Chang, "Robust multi-class transductive learning with graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 381–388.

[18] C. A. R. Sousa and G. E. A. P. A. Batista, "Robust multi-class graph transduction with higher order regularization," in *IJCNN*, 2015, pp. 299–306.

[19] R. Johnson and T. Zhang, "On the effectiveness of laplacian normalization for graph semi-supervised learning," *Journal of Machine Learning Research*, vol. 8, pp. 1489–1517, 2007.

[20] C. A. R. Sousa, S. O. Rezende, and G. E. A. P. A. Batista, "Influence of graph construction on semi-supervised learning," in *ECML/PKDD*, 2013, pp. 160–175.

[21] C. A. R. Sousa and G. E. A. P. A. Batista, "Constrained local and global consistency for semi-supervised learning," in *ICPR*, 2016.

[22] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: a geometric framework for learning from labeled and unlabeled examples," *Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.

[23] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer, 2009.

[24] T. Jebara, J. Wang, and S.-F. Chang, "Graph construction and b-matching for semi-supervised learning," in *ICML*, 2009, pp. 441–448.

[25] X. Zhou and M. Belkin, "Semi-supervised learning by higher order regularization," in *AISTATS*, 2011, pp. 892–900.

[26] S. Melacci and M. Belkin, "Laplacian support vector machines trained in the primal," *Journal of Machine Learning Research*, vol. 12, pp. 1149–1184, 2011.

[27] A. Goldberg, X. Zhu, and S. Wright, "Dissimilarity in graph-based semi-supervised classification," in *AISTATS*, 2007, pp. 155–162.

[28] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[29] C. A. R. Sousa, V. M. A. Souza, and G. E. A. P. A. Batista, "An experimental analysis on time series transductive classification on graphs," in *IJCNN*, 2015, pp. 307–314.