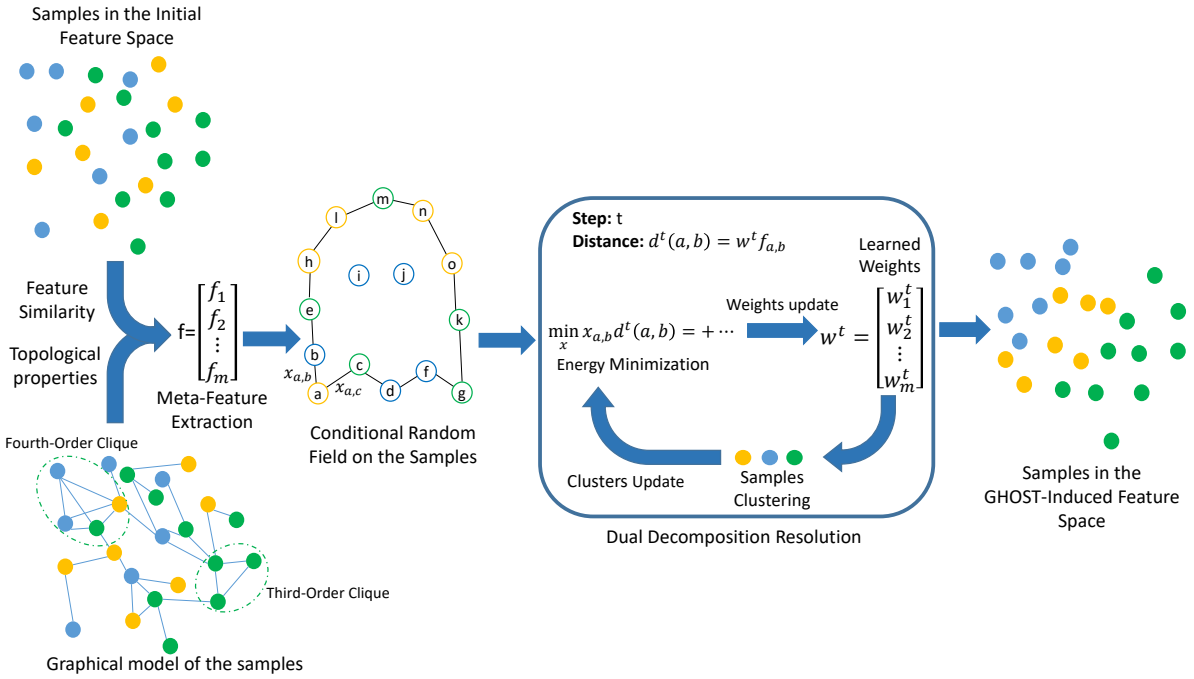
GHOST: Graph-based Higher-Order Similarity Transformation for Classification

Enzo Battistella, Maria Vakalopoulou, Nikos Paragios, Eric Deutsch

Graphical Abstract

GHOST: Graph-based Higher-Order Similarity Transformation for Classification

Enzo Battistella, Maria Vakalopoulou, Nikos Paragios, Éric Deutsch



Highlights

GHOST: Graph-based Higher-Order Similarity Transformation for Classification

Enzo Battistella, Maria Vakalopoulou, Nikos Paragios, Éric Deutsch

- Novel higher-order metric learning algorithm with Conditional Random Fields.
- Leverage graph and hyper-graph structures.
- Semi-supervised feature selection and weighting.
- Dedicated classification principle.

GHOST: Graph-based Higher-Order Similarity Transformation for Classification

Enzo Battistella[a,b,c], Maria Vakalopoulou[a,b], Nikos Paragios[a,b,d], Éric Deutsch[c]

[a] *Universite Paris-Saclay, CentraleSupelec, Mathematiques et Informatique pour la Complexite et les Systemes, Gif-sur-Yvette, France*
[b] *Universite Paris-Saclay, CentraleSupelec, Inria, Gif-sur-Yvette, France*
[c] *Universite Paris-Saclay, Institut Gustave Roussy, Inserm U1030 Molecular Radiotherapy and Innovative Therapeutics, Villejuif, France*
[d] *TheraPanacea, Paris, France*

Abstract

Exploring and identifying a good feature representation to describe high-dimensional datasets is a challenge of prime importance. However, plenty of feature selection techniques and distance metrics exist, which entails an intricacy for identifying the one best suited to the task. This paper provides an algorithm to design high-order distance metrics over a sparse selection of features dedicated to classification. Our approach is based on Conditional Random Field (CRF) energy minimization and Dual Decomposition, which allow efficiency and great flexibility in the considered features. The optimization technique ensures the tractability of high-dimensionality problems using hundreds of features and samples. Our approach is evaluated on synthetic data as well as on Covid-19 patient stratification. Comparisons with state-of-the-art baselines and our proposed method on different classification results prove the learned metric's relevance.

Keywords: Feature Selection, Distance Learning, Higher-order

1. Introduction

In machine learning, the choice of suitable feature spaces is of prime importance. Not only the dimensionality curse might affect the data, but some variables might be noisy or non-relevant. Many techniques have been investigated to select the most relevant features using some statistical metrics as correlation [1] or the importance weights an algorithm, as elastic-net, grants to the variables [2]. Also, many approaches, such as clustering algorithms, require a relevant metric to define a similarity notion between the samples. However, depending on the algorithm's mathematical properties and the metric, very different results are obtained [3]. To tackle the difficult task of designing a dedicated similarity function, some studies investigated semi-supervised clustering or distance learning [4, 5, 6]. Such approaches present the significant advantage of selecting the most relevant dimensions while warping the feature space, favoring the spatial proximity of samples presenting the same labels. Moreover, distance learning is semi-supervised. It aims at learning a good representation space, which can be learned from any clustering informative of the ground truth without the need for the actual labels. Despite the tremendous amount of work carried out on this topic, another aspect has been poorly considered, leveraging higher-order patterns in the data.

Graph structures are a standard model for data representation and allow great expressiveness. They have been adopted in various fields, from detecting network attacks [7] to modeling opinion diffusion processes [8], graph approaches' versatility is remarkable. They have already significantly spread to biological and medical data as they allow us to express better the intri-

cate interactions at stake [9]. Some of the prominent investigation areas are genomics [10] and spatial proteomics [11].

Moreover, many applications require leveraging the higher-order structures of a hypergraph [12, 13], a generalization of graphs where hyperedges can connect any number of nodes. However, higher-order information is generally computationally prohibitive to use in machine-learning-based methods[14]. Therefore, for tractability's sake, most techniques resort to surrogates such as casting a nonlinear hypergraph into a linear graph approximation [15], relying on small predefined patterns [16] to define a homogenous hypergraph where the hyperedges all have a fixed size, or summarising the graph structure by a feature such as clique order, centrality, or connectivity [17].

This study proposes a new approach to bridge the gap between feature selection, distance learning, and leveraging higher-order patterns. In particular, we propose a distance learning framework that can leverage complex higher-order information between the features. We define the notion of higher-order distance to characterize the similarity between groups of samples. To characterize the expressiveness of the learned feature space and compare it to other methods, we resort to classification and define a K-Nearest Neighbors classifier-based approach to leverage this higher-order metric on the samples. We report results on synthetic and medical datasets against state-of-the-art techniques, proving the superiority of our method.

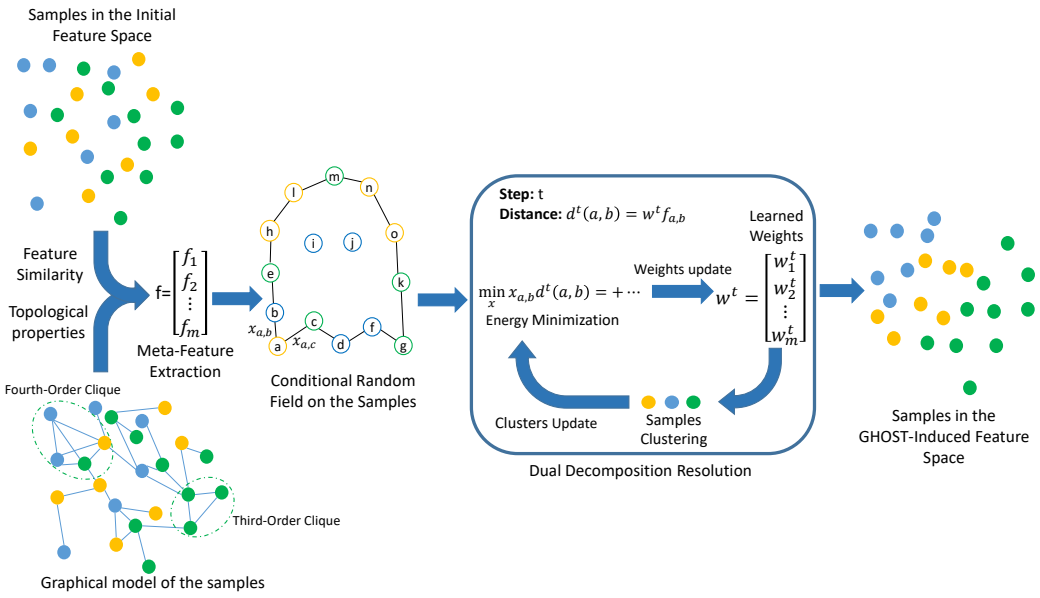


Figure 1: **Proposed GHOST approach.** A general overview of the different steps of our process. Our proposed GHOST approach aims at identifying the most relevant meta-features for a classification task. Meta-features might include any distance or higher-order metric between any number of the original features of the data. The approach relies on Conditional Random Field energy minimization for distance learning. It combines cluster label inference and optimization of meta-feature weights to induce a new metric space where samples with the same label are close.

2. Related Work

The main difficulty of clustering methods is to select the right data normalization, similarity notion and features to use [18]. Two main trends can be observed to tackle those issues, both leveraging contextual information. First, the constraint-based approaches consider partial annotations guiding the clustering process. Following this paradigm, the methods from [19] or [20] account for information on samples that must or cannot be clustered together to influence the clustering while being agnostic of the actual clustering. These

approaches might be further generalized to constraints on conjunctions and disjunctions of instances [21]. Second, metric learning approaches aim to learn a measure to discover specific properties. It offers the ability to identify structures similar to a given ground truth. This paradigm involves a more complex annotation as it requires actual clusterings. However, once the metric is learned, its strength dwells in its ability to be applied to other datasets without the need for any additional label. Besides, many studies demonstrated the essential role of the distance measure considered for clustering [5]. In this regard, in [22], the authors leverage a deep learning architecture to define a space representation, allowing them to define a better similarity notion between instances, while authors from [23] resort to a Support Vector Machine algorithm. In [4], constraints are defined to formulate the metric-learning task as a convex optimization problem.

Our study is based on the work of [6], which presents the advantage of allowing a general definition of distance, performing simultaneously feature and metric selection as well as their weighting. Besides, it is designed for a center-based clustering algorithm, of which K-Means is a popular example. This formulation presents the asset of defining a relevant cluster representative whose good properties have been demonstrated for feature selection in [24, 3]. This clustering paradigm is the critical point in our approach for exploiting higher-order graph structures in a reasonable computational time and space. This approach relies on the expression power of CRF.

Despite CRF's excellent expressiveness and ability to capture higher-order relations, the toll for fully leveraging this higher-order information might be heavy on memory and time consumption. To cope with this issue, authors

from [25, 26] proposed to exploit the binary nature of the CRF labels in our settings to optimize the resolution. Finally, in [27], dual decomposition is exploited to divide the initial energy to minimize in several easier-to-solve sub-problems.

In this study, we adapt and extend the formulation of metric learning presented in [6]. The main contribution is to bring center-based clustering and the notion of pairwise metrics to higher-order settings through the inclusion of graph properties. In addition, we modified the error function originally considered to better account for unbalanced classes. Finally, we propose an approach to leverage graph structural information from our data.

3. Methodology

Without loss of generality, let us define a metric assessing the similarity of a set of h objects as a h^{th}-order metric. For instance, a usual distance is referred to as a 2^{nd}-order metric or pairwise metric. For the simple case of pairwise metrics, we based our study on [6]. We extend this formulation by introducing h^{th}-order metrics for any $h > 2$. The case of 3^{rd}-order metrics will be more detailed for clarity's sake. However, the same approach is applied to obtain the results for any order. The proposed GHOST approach is represented in Figure 1.

3.1. Center-Based Clustering

As in [6], our approach is based on center-based clustering. Considering a set of objects to cluster $\mathcal{V}$, we define a set of binary variables $\{x_{p,q}\}_{p,q \in V}$ indicating whether p is assigned to the cluster of center q, $x_{p,q} = 1$, or not,

$x_{p,q} = 0$. We consider a distance $d_{p,q}$ between objects p and q.

$$\begin{aligned} &\min_x \sum_{p,q \in V} d_{p,q} x_{p,q} \quad s.t. \ \sum_{q \in V} x_{p,q} = 1, \ \forall p \\ &x_{p,q} \leq x_{q,q}, \quad x_{p,q} \in \{0,1\}, \ \ \forall p, q. \end{aligned} \tag{1}$$

The above system minimizes, with respect to three constraints, the distance between a point and the center of the cluster it is assigned to. First, each point has to belong to one and only one cluster. Second, if a point is assigned to a cluster center, this center must be assigned to itself. Third, assignment variables are binary. [6] casts the previous optimization problem as an equivalent energy minimization task:

$$E(x, d) = \sum_{p,q} u_{p,q}(x_{p,q}, d) + \sum_{p,q} \phi_{p,q}(x_{p,q}, x_{q,q}) + \sum_{p} \phi_p(x_p) \tag{2}$$

$u_{p,q}$ being the second-order potentials of the CRF standing for the distance to the cluster center and ϕ_p, $\phi_{p,q}$ the constraints. $u_{p,q}$ can be defined as the effective distance between p and its current cluster center, as it will be null for any q different from the center. More precisely:

$$\begin{aligned} &u_{p,q}(x_{p,q}, d) = d_{p,q} x_{p,q} \\ &\phi_{p,q}(x_{p,q}, x_{q,q}) = \delta(x_{p,q} \leq x_{q,q}), \quad \phi_p(x_p) = \delta(\sum_q x_{p,q} = 1) \end{aligned} \tag{3}$$

with $x_p = \{x_{p,q} \mid q \in V\}$ and $\delta(e) = 0$ if e True and ∞ otherwise.

This study defines a generalization of this energy formulation for clustering. Here is an example in a third-order setting. In addition to the previous notations, we consider a third-order distance $d_{p,p',q}$ for any triplet of objects to cluster p, p', and q. Examples of such higher-order distances are given in

section 3.10.

$$E(x,d) = \sum_{p,q} u_{p,q}(x_{p,q},d) + \sum_{p,p',q} u_{p,p',q}(x_{p,q}x_{p',q},d) + \sum_{p,q} \phi_{p,q}(x_{p,q},d_{q,q}) + \sum_{p} \phi_p(x_p) \tag{4}$$

the new function $u_{p,p',q}$ being the third-order potentials of the CRF, everything else remaining unchanged. More precisely:

$$u_{p,p',q}(x_{p,q}x_{p',q},d) = d_{p,p',q}x_{p,q}x_{p',q} \tag{5}$$

3.2. Metric Learning Formulation

Our framework learns a distance between objects using a set of K training subjects $\{V^k, \mathcal{C}^k, y^k\}$ for each set $k \in K$, V^k is the set of objects to be clustered according to ground truth $\mathcal{C}^k$ and knowing input data y^k. We are also assuming that we can get from the input data a positive feature function for each pair of objects p, q as $f_{p,q}(y^k)$ and for each triplet of objects p, p', q as $f_{p,p',q}(y^k)$. The codomain of the feature's functions will be called meta-feature space as it is obtained from the task's actual features and is GHOST's input space. We consider a meta-feature space of size d. One should notice that even though there is a ground truth cluster for each set, the cluster centers remain unknown. A feasible solution x^k of $\mathcal{C}^k$ denoted as $x^k \in \mathcal{X}(\mathcal{C}^k)$, will consist in a set of assignments such as for each ground truth cluster $C \in \mathcal{C}^k$ all the objects $p \in C$ are assigned to the same center $q \in C$. Besides, we are looking for a distance over a set S of cardinal $2 \leq \mid S \mid \leq 3$ expressed as:

$$d_S^k = \begin{cases} d_{p,q}^k = w^T f_{p,q}(y^k) \ if \ S = \{p,q\} \\ d_{p,p',q}^k = w^T f_{p,p',q}(y^k) \ otherwise \end{cases} \tag{6}$$

For conciseness sake's, we will denote $E^k(x,d) = E(x,d^k)$ and $u^k(x,d) = u(x,d^k)$.

w **being the weight vector we want to estimate.** At the difference of the formulation proposed in [6], we impose $w_i \geq 0$, $\forall i \leq d$. This specificity aims to enforce the positivity of the distance obtained. Also, as it has been presented in [6], a projection of the weights onto $\mathbb{R}_+$ ensures better performance in the second-order settings. We impose this constraint to improve the higher-order distance learning resolution's tractability, as highlighted in the proofs (in Appendix).

Notice that GHOST is very robust and flexible as we use the same weight vector w for both the second-order and the third-order distances, which means that a component i of vectors $f_{p,q}(y^k)$ and $f_{p,p',q}(y^k)$ have to relate to the same property. The i-th component of $f_{p,p',q}(y^k)$ can be a generalization of $f_{p,q}(y^k)$, but it can also stand alone, and in this case, $f_{p,q}(y^k)$ will be null. Similarly, a component $f_{p,q}(y^k)$ might not possess any relevant generalization, and in this case, $f_{p,p',q}(y^k)$ will be null. For instance, a suitable third-order function $f_{p,p',q}$ could have for component the perimeter or surface of the triangle $\{p,p',q\}$. With this particular example, in addition to the initial second-order warping of the space, such as having a small distance between each object of the cluster and the center, we will also have a small distance between pairs of objects. However, much more intricate properties can be introduced. For instance, we can consider statistical distances as the Mahalanobis distance between the set of observations $\{p,p'\}$ and q, which is designed to estimate if the object q is a natural center for the set $\{p,p'\}$ regarding mean and variance considerations. We will present in Section 3.10,

possible higher-order feature functions definition on graph structures.

As in [6], a Max-Margin approach is considered to approximate w. We are looking for $x^k \in \mathcal{X}(\mathcal{C}^k)$ whose energy $E^k(x^k, d)$ is smaller than the energy of any other solution x by an error function $\Delta(x, d)$ to be defined, i.e.

$$\exists x^k \in \mathcal{X}(\mathcal{C}^k),\ E^k(x^k, d) \leq E^k(x, d) - \Delta(x, d) + \xi_k \tag{7}$$

where slack variable ξ_k is considered in the case of infeasible training sets. Adding this constraint to the previous energy minimization problem gives the regularized loss:

$$\min_{\{x^k \in \mathcal{X}(\mathcal{C}^k)\}} \tau J(w) + \sum_k \mathcal{L}_{E^k} \tag{8}$$

where $J(w)$ is a regularization term penalizing w complexity e.g. lasso regularization, while the hinge loss $\mathcal{L}_{E^k}$ includes ξ_k and is expressed as:

$$\mathcal{L}_{E^k}(x^k, w) = E^k(x^k, w) - \min_x (E^k(x, w) - \Delta(x, \mathcal{C}^k)) \tag{9}$$

It favors feasible solutions with energy close to the minimal energy for any possible assignment penalized by the error function according to the violated constraints.

3.3. Max-Margin Energy

The good choice of $\Delta(x, \mathcal{C}^k)$ is essential to obtain a relevant w. In particular, we need this error function to be 0 if $x \in \mathcal{X}(\mathcal{C}^k)$ and to have a value representing on what extent x violates the constraints imposed by the ground truth $\mathcal{X}(\mathcal{C}^k)$. We consider the training set k with cluster ground truth $\mathcal{C}^k$ the function:

$$\Delta(x, C^k) = \alpha \sum_{C \in \mathcal{C}^k} W(1 - \sum_{q \in C} x_{q,q}) + \beta \sum_{C \in \mathcal{C}^k} \frac{1}{\mid C \mid} \sum_{p \in C} (1 - \sum_{q \in C} x_{p,q}) \tag{10}$$

with $W(z) = |z|([z < 0].(\mid V^k \mid - \mid C \mid) + [z > 0]. \mid C \mid)$, [.] being the indicator function. The first term penalizes solutions x presenting no or several exemplars for a ground truth cluster $C \in \mathcal{C}^k$. We have adapted the error function W proposed in [6] to better account for unbalanced classes with an additional penalty on the sizable clusters presenting no exemplars or the small clusters presenting several exemplars. If there is more than one center in C, W will increase with the number of centers and the difference between the size of V^k and C. If there is no center in C, W will increase with the size of C. The second term penalizes the solutions that do not assign an object of a ground truth cluster $C \in \mathcal{C}^k$ to an exemplar from C. We considered a weight inversely proportional to the cluster's size to balance the importance of small clusters in the learning process. The learning constants α and β characterize the relative importance of having a center per cluster and assigning all objects to a center from their ground truth cluster. If we are confident in the number of clusters, if they present a contextual significance, for instance, but the separation between the two clusters is fuzzy, we would set $\alpha > \beta$. If we want to maintain the separation between the known clusters but the actual number of clusters is unclear, we would set $\beta > \alpha$. In the following, we want to maintain both aspects equally, so we consider $\alpha = \beta$. The regularized loss defined in equation 8 can be expressed as a new CRF energy $\bar{E}^k = E^k - \Delta$:

$$\begin{aligned}\bar{E}^k(x, w) = \sum_{p,q} \bar{u}^k_{p,q}(x_{p,q}) + \sum_{p,p',q} \bar{u}^k_{p,p',q}(x_{p,q}x_{p',q}) + \\ \sum_{p,q} \bar{\phi}_{p,q}(x_{p,q}) + \sum_{p} \bar{\phi}_{p}(x_{p}) + \sum_{C \in \mathcal{C}^k} \bar{\phi}_{C}(x_C) - \beta|\mathcal{C}^k|\end{aligned} \tag{11}$$

with:

$$\begin{aligned}
&\bar{u}^k_{p,q}(x_{p,q}) = u^k_{p,q}(x_{p,q}, d) + \beta[\exists C \in \mathcal{C}^k,\ p, q \in C]\frac{x^k_{p,q}}{\mid C \mid} \\
&\bar{u}^k_{p,p',q}(x_{p,q}x_{p',q}) = u^k_{p,p',q}(x_{p,q}x_{p',q}, d) \\
&\bar{\phi}_{p,q}(x_{p,q}) = \phi_{p,q}(x_{p,q}, x_{q,q}) \\
&\bar{\phi}_p(x_p) = \phi_p(x_p),\ x_p = \{x_{p,q} \mid q \in V\} \\
&\bar{\phi}_C(x_C) = -\alpha W(1 - \sum_{q \in C} x_{q,q})
\end{aligned} \tag{12}$$

It is interesting to notice that thanks to the property of Δ for a feasible solution $x^k \in \mathcal{X}(\mathcal{C}^k)$, $\bar{E}^k(x^k, w) = E^k(x^k, w)$.

3.4. Optimizing over $\{x^k\}$

For a fixed w, minimizing $\bar{E}^k(x^k, w)$ requires the constraints to be satisfied, $\bar{\phi}_p(x^k_p) = 0$ and $\bar{\phi}_{p,q}(x^k_{p,q}) = 0$, which entails $x^k \in \mathcal{X}(C^k)$. And, in this case, $\Delta(x, C^k) = 0$. Thus,

$$x^k = \underset{x \in \mathcal{X}(C^k)}{\arg\min}(\sum_{p,p',q} d_{p,p',q}x_{p,q}x_{p',q} + \sum_{p,q} d_{p,q}x_{p,q}) \tag{13}$$

To minimize this problem, we only need to find the set Q^k of exemplars q minimizing the above function per cluster in $\mathcal{C}^k$ and then assign each point of the cluster to its exemplar. In this case, the constraints will be satisfied as we ensure each cluster has one and only one center and assign all cluster samples to this center.

3.5. Dual Decomposition

A dual decomposition approach is a widespread approach in optimization. Its efficient resolution by projected subgradients has been introduced for

MRF and CRF in [28]. It presents the crucial property of optimally solving the dual linear programming problem. Besides, its versatility allows the generalization to higher-order CRF as performed in [27].

The dual decomposition principle aims at isolating several much easier subproblems tailored to be equivalent to the original problem after summation. A simple and popular decomposition is to consider each node independently. Each subproblem might be solved by very efficient inference techniques as graph-cuts approaches, without venturing into a scaling issue. The global resolution leads to a projected subgradient scheme, provably offering an optimal solution. Formally, dual decomposition expression relies on simple subproblems, also called slave problems, and on a master problem enacting as a coordinator.

Here, for each $k < K$, we define a slave problem per datapoint $p \in V^k$, $\bar{E}^k_p$, and one per cluster $C \in \mathcal{C}^k$, $\bar{E}^k_C$.

$$\begin{aligned}
\bar{E}^k_p(x,w) =& \sum_{q \neq p} \bar{u}^k_{p,q}(x_{p,q}) + \sum_{p',q \neq p} \bar{u}^k_{p,p',q}(x_{p,q}x_{p',q}) + \\
& \sum_q \bar{\phi}_{p,q}(x_{p,q}) + \bar{\phi}_p(x_p) - \frac{\beta}{\mid V^k \mid} + \\
& \sum_q \Big(\frac{1}{\mid V^k \mid +1}\big(\bar{u}^k_{p,q}(x^k_{q,q}) + \sum_{p'} \bar{u}^k_{q,p',q}(x^k_{q,q}x^k_{p',q})\big) + \lambda_{p,q}x_{q,q}\Big) \qquad (14) \\
\bar{E}^k_C(x,w) =& \bar{\phi}_C(x_C) + \sum_q \Big(\frac{1}{\mid V^k \mid +1}\big(\bar{u}^k_{p,q}(x^k_{q,q}) + \\
& \sum_{p'} \bar{u}^k_{q,p',q}(x^k_{q,q}x^k_{p',q})\big) + \lambda_{C_q}x_{q,q}\Big)
\end{aligned}$$

where the Lagrangian variables $\lambda = \{\{\lambda_{p,q}\}, \{\lambda_{C_q}\}\}$ are used to ensure the consistency of the solution. We impose the satisfaction of: $\lambda \in \Lambda^k = \{\lambda : \sum_{p \in S^k} \lambda_{p,q} + \lambda_{C_q} = 0, \forall C \in \mathcal{C}^k, q \in C\}$. Therefore, by design, $\bar{E}^k(x^k, w) =$

$\sum_p \bar{E}_p^k(x,w) + \sum_C \bar{E}_C^k(x,w)$. Thus, using the hinge loss define in equation 9 per subproblem, the loss function to be minimized is:

$$\min_{\{x^k \in \mathcal{X}(\mathcal{C}^k)\}, w, \{\lambda^k \in \Lambda^k\}} \tau J(w) + \sum_k \sum_{p \in V^k} \mathcal{L}_{\bar{E}_p^k} + \sum_k \sum_{C \in \mathcal{C}^k} \mathcal{L}_{\bar{E}_C^k} \tag{15}$$

3.6. Slave Problems Optimization

To optimize w, we first need to solve the slave problems by leveraging their specific structures. An essential characteristic to notice is that, for fixed $\{x^k\}$, the slaves' energy can be related to CRF energies. Details of all the proofs and computation steps are provided in Supplementary Materials.

3.6.1. Point-Wise Subproblem Solution $\{\hat{x}^{k,p}\}$

Regarding the point-wise subproblems, we proceed as follows. The solution in pairwise settings has been demonstrated in [6]. In the following lemma, we generalize the solution that was proposed to a third-order context as:

Lemma 1. *For fixed $p \in V^k$, let $\theta_q^k = \dfrac{\bar{u}_{q,q}^k(1) + \bar{u}_{q,q,q}^k(1)}{|V|^k + 1} + \lambda_{p,q}$ and $\bar{\theta}_q^k = [\theta_q^k]_+ + \bar{u}_{p,q}^k(1) + \bar{u}_{p,q,q}^k(1) + \bar{u}_{p,p,q}^k(1)$ where $[z]_+ = max(0,z)$. minimizer $\hat{x}^p$ of $\bar{E}_p^k(x,w,\lambda^k)$ is given by*

$$\hat{x}_{q,q}^p = [\theta_q^k < 0], \quad \hat{x}_{p,q}^p = [q = \bar{q}] \text{ where } \bar{q} = \arg\min_q(\bar{\theta}_q^k) \tag{16}$$

3.6.2. Cluster-Wise Subproblem Solution $\{\hat{x}^{k,C}\}$

Regarding the cluster-wise subproblems, we proceed as follows. The solution in pairwise settings has been demonstrated in [6]. We can notice that our formulation of the cluster-wise subproblem presents a high similarity with the original formulation, and the only difference for the optimization is in θ_q^k expression:

Lemma 2. *For fixed $C \in \mathcal{C}^k$, let $\theta_q^k = \frac{\bar{u}_{q,q}^k(1) + \bar{u}_{q,q,q}^k(1)}{\mid V^k \mid +1} + \lambda_{C_q}$, $\forall q \in C$. A minimizer $\hat{x}^C$ of $\bar{E}_C^k(x, w, \lambda^k)$ is given, $\forall q \in C$, by*

$$\hat{x}_{q,q}^C = \begin{cases} [\theta_q^k < \alpha(\mid V^k \mid - \mid C^k \mid)], \\ \quad if \ \sum_{q' \in C} [\theta_q^k - \alpha(\mid V^k \mid - \mid C^k \mid)]_- + \alpha \mid V^k \mid < 0 \\ 0 \ otherwise \end{cases} \tag{17}$$

with $[z]_- = min(0, z)$.

3.6.3. Optimizing over λ and w

To optimize over λ and w, we perform an iterative projected subgradient approach:

$$w \longleftarrow w - s_t \delta_w, \quad \lambda^k \longleftarrow proj_{\Lambda^k}(\lambda^k - s_t \delta_{\lambda_k}) \tag{18}$$

with $\{\delta_w\}$ and $\{\delta_{\lambda_k}\}$ subgradient functions and $proj_{\Lambda^k}$ the projection onto Λ^k. Then, the following lemma gives the updates to be applied iteratively to efficiently obtain the approximation of w and $\{\lambda_{p,q}^k, \lambda_{C_q}^k\}$. The updates are obtained by summing the respective updates of each subproblem according to the formula provided in equation 15.

Lemma 3. *Let s_t be the weight granted to the optimization at step t. We define $\hat{X}_q^k = \hat{x}_{q,q}^{k,C} + \sum_p \hat{x}_{q,q}^{k,p}$ and $\hat{X}_{p,q}^k = \hat{x}_{q,q}^{k,p} \hat{x}_{p,q}^{k,p} + \hat{x}_{q,q}^{k,C}[p = q, q \in C]$. Then, update reduces to:*

$$\begin{aligned} w \ -&= s_t(\tau \nabla J(w) + \sum_k \delta_w^k), \quad \lambda_{p,q}^k \ -= s_t(\frac{\hat{X}_q^k}{\mid V^k \mid +1} - \hat{x}_{q,q}^{k,p}) \\ \lambda_{C_q}^k \ -&= s_t(\frac{\hat{X}_q^k}{\mid V^k \mid +1} - \hat{x}_{q,q}^{k,C}) \end{aligned} \tag{19}$$

where

$$\delta_w^k = \sum_{p,p',q \in V^k} x_{p,q}^k x_{p'q}^k f_{p,p',q}^k + \sum_{p,q \in V^k} x_{p,q}^k f_{p,q}^k - (\sum_{p,q \neq p \in V^k} (\hat{x}_{p,q}^{k,p} \hat{x}_{q,q}^{k,p} f_{p,q,q}^k + \hat{x}_{p,q}^{k,p} f_{p,p,q}^k) + \sum_{p,q \neq p \in V^k} \hat{x}_{p,q}^{k,p} f_{p,q}^k + \sum_{q \in V^k} \frac{1}{\mid V^k \mid +1} (\hat{X}_q^k f_{q,q}^k + \sum_{p \in V^k} \hat{X}_{p,q}^k f_{q,p,q}^k)) \tag{20}$$

Note that $\nabla J(w)$ has to refer to a subgradient if J is non-differentiable. Besides, a constraint can be imposed over w by applying a projection during w update.

$$w \longleftarrow proj_W(w - s_t \delta_w) \tag{21}$$

where W can be any convex set of constraints included in $\mathbb{R}_+$. For instance, we mimic a true distance and satisfy the positivity constraint over w with $proj_W(z) = max(z, 0)$. It will also enforce an additional sparsity on the weight vector. Notice that at least a positivity constraint has to be imposed for the resolution of the higher-order distance learning. We summarize the complete learning process in Algorithm 1.

To improve the tractability of the approach, we leveraged a stochastic gradient descent (sgd) framework. It consists in randomly selecting a subset of the training samples at each subgradient iteration and performing the updates by relying only on those samples.

3.7. Generalization to Higher-Order Distances

Our approach's strength is its ability to be efficiently generalized to any order. Let $h \geq 2$ be the order of the distance we are looking for. h corresponds to the maximal set size we will consider in our metric definition. Our target metric considers any set S of size $\mid S \mid \leq h$ and is defined as

Algorithm 1: Learning Process

Data: training cohorts $\{V^k, \mathcal{C}^k, y^k\}$, features functions $\{f^2_{p,q}(y^k), f^3_{p,p',q}(y^k)\}$

$\lambda^k \leftarrow 0,\ \forall k$

do

  Optimize x^k: $\forall C \in \mathcal{C}^k$,

  $q_c = arg\min_{q \in C}(\sum_{p,p' \in C} d_{p,p',q} x_{p,q} x_{p'q} + \sum_{p \in C} d_{p,q} x_{p,q})$;

  $x^k_{p,q} = 1,\ p \in C \iff q = q_C$;

  Iterate T subgradient updates:

  repeat

    Solve slaves $\bar{E}^k_p$, $\bar{E}^k_C$ via lemmas 1, 2;

    Update w, λ^k via lemma 3

  until *T times*;

  Project w over $W \subset \mathbb{R}_+$

while *Not Convergence*;

$d_S = w^T f_{\{p\}_{p\in S}}(y)$ where $f_{\{p\}_{p\in S}}$ is a positive feature function providing a closeness score on set S. This metric aims to establish a characterization of the meaningfulness of samples grouped in S altogether.

A formal mathematical description is provided in Appendix D.

3.8. Extension to Cluster Metrics

A final interesting addition we can bring to our higher-order distance learning framework is to consider a metric between a sample and a ground truth cluster. The difference between this particular setting and the previous higher-order metrics is that here, we will consider a metric able to tackle sets of objects of different sizes (the size of the clusters) and thus will not have a defined order. The interest in such a distance is to benefit from a structural metric characterizing the closeness between a sample and a given cluster. During inference, it will be especially valuable for identifying the most suited cluster for a sample.

A formal mathematical description is provided in Appendix E.

3.9. Time Complexity Analysis

The time complexity of one update of this learning process is $O(d\sum_{k=1}^{K}(\frac{|V^k|}{T})^h)$ where h is the maximal order considered. The division by T is introduced thanks to the sgd formulation. The complexity of one update in the original second-order formulation [6] was $O(d\sum_{k=1}^{K}|V^k|^2)$. The impact on the runtime of going from the second order to the third order is presented in Section 5.

Without balanced error	**Balanced Accuracy**		**Weighted Precision**		**Weighted Sensitivity**		**Weighted Specificity**	
Without path feature	**Validation**	**Test**	**Validation**	**Test**	**Validation**	**Test**	**Validation**	**Test**
Average	0.6	0.42	0.76	0.35	0.62	0.4	0.58	0.43
Min	0.52	0.4	0.75	0.27	0.54	0.38	0.49	0.42
Min Max	0.59	0.38	0.68	0.32	0.63	0.37	0.6	0.39
Min center	0.62	0.34	0.75	0.26	0.6	0.32	0.57	0.35
KNN	0.62	0.41	0.69	0.29	0.64	0.4	0.61	0.43
With balanced error	**Balanced Accuracy**		**Weighted Precision**		**Weighted Sensitivity**		**Weighted Specificity**	
Without path feature	**Validation**	**Test**	**Validation**	**Test**	**Validation**	**Test**	**Validation**	**Test**
Average	0.71	0.61	0.72	0.62	0.72	0.6	0.71	0.62
Min	0.66	0.62	0.68	0.62	0.67	0.62	0.65	0.63
Min Max	0.65	0.58	0.65	0.59	0.65	0.56	0.65	0.59
Min center	0.74	0.62	0.75	0.64	0.75	0.61	0.74	0.63
KNN	0.72	0.6	0.73	0.62	0.72	0.58	0.73	0.61
With balanced error	**Balanced Accuracy**		**Weighted Precision**		**Weighted Sensitivity**		**Weighted Specificity**	
With path feature	**Validation**	**Test**	**Validation**	**Test**	**Validation**	**Test**	**Validation**	**Test**
Average	1	0.77	1	0.84	1	0.73	1	0.82
Min	1	0.76	1	0.83	1	0.71	1	0.81
Min Max	1	0.79	1	0.85	1	0.75	1	0.83
Min center	1	0.78	1	0.8	1	0.73	1	0.8
KNN	1	0.69	1	0.81	1	0.63	1	0.75

Table 1: Results with the synthetic dataset of the different experiments in the second-order settings for the various inference strategies. Training results are not reported as reaching 1 in all cases.

3.10. Extracting and Leveraging Structural Information from Data

Several approaches exist in order to design a graph structure on data sets with no natural graph representation. Here, we relied on a distance matrix between objects computed as the sum of all the different feature functions used in distance learning. Then, a k-nearest neighbors approach was computed, meaning that there is an edge between two objects p and q iff p (resp. q) is in the k objects the closest of q (resp. p).

Once a graph structure was obtained, we studied different ways of leveraging their properties. Our first, most simple approach is considering the

shortest path $S_{p,q}$ between objects p and q in the graph. The distance between those objects will then be the weighted length $L_{p,q}$ of such a path. The generalization of this method for a set of objects $\{p_1, ..., p_l\}$ and a potential center q is defined as follows:

$$SP_l(p_1, ..., p_l) = \sum_{i \in \{1,...,l\}} \sum_{j \in \{i+1,...,l\}} \frac{L_{p_i,q} + L_{p_j,q}}{L_{p_i,p_j}}$$

The interpretation of this graph metric is that the center of a cluster q has to be a hub for the objects of its cluster i.e. the ratio between the shortest path and the shortest path passing by q has to be small for any pair of objects in the cluster.

Similarly, we considered the eccentricity of a set of objects $\{p_1, ..., p_l\}$ as a l-order graph metric. We deem a set of objects to have a small maximal weighted graph diameter to belong to the same cluster.

Then, we considered two connectivity-related metrics. The first one is based on the clique order of a set of objects $CO(p_1, ..., p_l)$ and is defined as $max\_degree(G) - CO(p_1, ..., p_l)$ with $max\_degree(G)$ the maximal degree of a node in the whole graph. By doing so, we consider that the bigger the clique order in the set of objects, the more relevant their association in a cluster.

The second metric is based on the connectivity resilience $CR(p_1, ..., p_l)$, which is the minimal number of nodes to remove to disconnect the set of objects. The metric is defined as $l - CR(p_1, ..., p_l)$.

3.11. Leveraging a Task Dedicated Distance for Classification

In order to perform the classification, we relied on a K-Nearest Neighbors framework. Once the distance was learned, we labeled a new sample as the

ground truth cluster from which it is the closest. We experimented and compared different strategies to determine the closest cluster: 1) Average distance to the points of the cluster, 2) Minimum distance to the points of the cluster, 3) Maximum distance to the points of the cluster, 4) Min center: Distance to the center of the cluster, 5) Majoritarian cluster of the k-nearest neighbors. The distance between the new sample p and objects of the cluster C is computed using the learned dedicated distance. For $l > 2$-order distances, we compute the distance on the set $\{p, p_1, ..., p_{l-2}, q\}$ where we iterate over all possible sets $\{p_1, ..., p_{l-2}\} \in C^{l-2}$ and q is the cluster center discovered during the learning step.

4. Implementation Details

We implemented the algorithm proposed in [6] and used it as a baseline. It is available at `https://github.com/ebattistella/Second-order-Distance-learning`. Besides, the adaptation to general higher-order distances we propose in this study is available at `https://github.com/ebattistella/Higher-order-Distance-Learning-GHOST-`.

To prove the relevance of our higher-order formulation, we leveraged two datasets of very different natures. First, we synthesized a dataset with samples in dimension 100 with 60 noisy dimensions. Clusters are designed by considering 100 samples generated from Gaussian distributions with different variances and means between the two clusters on the non-noisy dimensions. The noise is simulated by taking a much larger variance. Ideal graphs were generated on this dataset as one clique per cluster with no connection between cliques. Then, we added noise to the graphs using a rewiring method [29].

For each pair of nodes, we added or removed an edge with a probability of p. We considered values of $p \in [0, 0.5]$ with an increment of 0.1. We generated training, validation, and test sets considering different variances and means. For each set, we considered base variances randomly chosen for each feature between 0 and 200 shifted by respectively 10, 30, and 50 for the non-noisy dimensions and 1000, 2000, and 10000 for the noisy ones. Regarding the means, we considered base means randomly chosen for each feature between -50 and 50 shifted by 0, 10, and 50, respectively. The aim was here to visualize the generalizability of the learned weights and their resilience to increasing noise. We applied the same principle and distributions to generate a simulated benchmark presenting 3 clusters and 100 samples. We then leveraged a Covid-19 dataset introduced in [30]. We used the same training and testing sets as the authors and compared their results to the classification performance of our proposed GHOST approach over the Severe/Non-severe staging task. The two classes are defined as the several cases, patients needing intubation after four days of the onset of the disease, and the non-severe cases, patients recovering after four days of the disease onset. A graph of the data was obtained through the method proposed in Section 3.10 by considering the 5 closest neighbors.

The second-order feature functions we based all our experiments on are feature-wise Euclidean distances. In addition, to assess the influence of different metrics, we considered meta-features Euclidean, Minkowski, City-block, Cosine, Correlation, Hamming, Jaccard, Chebyshev, Matching, Yule, Braycurtis, Dice, Kulsinski, Russellrao, Pearson-correlation based, Spearman-correlation based, Kendall-correlation based distances on the full feature

space.

We first used the simple pairwise distance defined using the basic formulation from [6]. Then, we complemented this with our balanced error function to better account for the cluster size. We finally added a shortest-paths-based metric SP_2 to assess the value of graph information even in second-order settings and compare it with the higher-order. We performed the higher-order distance learning using the combination of the second-order meta-features with the different higher-order metrics defined in Section 3.10. We considered the third-order approach, the cluster metric approach, and the combination of both approaches, which constitutes our GHOST approach.

w is initialized with ones on all dimensions.

5. Results and Discussion

5.1. Synthesized Dataset

In this subsection, we considered the two synthesized clusters. This experiment aimed to assess the capacity of our higher-order framework to leverage information from a graph according to its level of noise and combine it with usual second-order metrics to perform classification. We used as a baseline the second-order framework performances with and without considering path length information in a graph. The assessment metrics used are described in Appendix F. First, we reported the results in the second-order without graph information nor balanced error function, without graph information and with balanced error function, and with both in Table 1. Here, we reported the performance of the different strategies to infer the label of a new sample defined in Section 3.11. We observed superior results of the distance to cluster

Third-order only	**Balanced Accuracy**			**Weighted Precision**			**Weighted Sensitivity**			**Weighted Specificity**		
p	**Training**	**Validation**	**Test**	**Training**	**Validation**	**Test**	**Training**	**Validation**	**Test**	**Training**	**Validation**	**Test**
0	1	1	1	1	1	1	1	1	1	1	1	1
0.1	0.97	0.91	0.96	0.97	0.92	0.96	0.97	0.92	0.96	0.97	0.91	0.96
0.2	0.9	0.92	0.91	0.9	0.93	0.92	0.89	0.92	0.9	0.9	0.93	0.92
0.3	0.59	0.58	0.64	0.78	0.77	0.79	0.61	0.6	0.61	0.63	0.62	0.67
0.4	0.62	0.62	0.68	0.78	0.78	0.8	0.59	0.58	0.65	0.66	0.65	0.71
0.5	0.54	0.52	0.48	0.57	0.54	0.48	0.51	0.5	0.46	0.57	0.55	0.5
Cluster metric only	**Balanced Accuracy**			**Weighted Precision**			**Weighted Sensitivity**			**Weighted Specificity**		
p	**Training**	**Validation**	**Test**	**Training**	**Validation**	**Test**	**Training**	**Validation**	**Test**	**Training**	**Validation**	**Test**
0	1	1	1	1	1	1	1	1	1	1	1	1
0.1	0.97	0.91	0.96	0.97	0.92	0.96	0.97	0.92	0.96	0.97	0.91	0.96
0.2	0.9	0.92	0.91	0.9	0.93	0.92	0.89	0.92	0.9	0.9	0.93	0.92
0.3	0.7	0.66	0.64	0.78	0.77	0.79	0.7	0.63	0.61	0.63	0.62	0.67
0.4	0.61	0.6	0.63	0.76	0.75	0.8	0.6	0.59	0.6	0.67	0.66	0.7
0.5	0.53	0.48	0.5	0.55	0.47	0.5	0.5	0.47	0.46	0.5	0.49	0.5
GHOST	**Balanced Accuracy**			**Weighted Precision**			**Weighted Sensitivity**			**Weighted Specificity**		
p	**Training**	**Validation**	**Test**	**Training**	**Validation**	**Test**	**Training**	**Validation**	**Test**	**Training**	**Validation**	**Test**
0	1	1	1	1	1	1	1	1	1	1	1	1
0.1	0.98	0.91	0.97	0.97	0.92	0.96	0.98	0.92	0.95	0.98	0.91	0.97
0.2	0.91	0.93	0.91	0.9	0.93	0.92	0.89	0.92	0.9	0.9	0.93	0.92
0.3	0.71	0.65	0.65	0.79	0.76	0.8	0.71	0.63	0.62	0.62	0.61	0.68
0.4	0.62	0.62	0.68	0.78	0.78	0.8	0.6	0.59	0.6	0.67	0.66	0.7
0.5	0.54	0.52	0.48	0.57	0.54	0.48	0.51	0.5	0.46	0.57	0.55	0.5

Table 2: Results on the synthetic dataset of the different higher-order experiments with the distance to center inference strategy.

center approach on all metrics. This trend was consistent in the different experiments we performed. Thus, in the following, we only reported results from this strategy for concision's sake. In addition, it is interesting to notice that using the class balanced error function as described in 3.2 significantly increases the performance from random results (< 0.5 on Test) to results > 0.6. Finally, the graph information is highly valuable here as it brings the results over all the metrics above 0.7.

Table 2 presents the comparison of the performance using the different frameworks defined in this study. The second-order results reported in Ta-

ble 1 with path length information have been obtained with the ideal graph without noise. The foremost point to notice is the greater performance and lesser overfitting of the methods leveraging third-order information compared to the second-order. Indeed, the results of the higher-order approach with noise up to 40% (rewiring probability up to 0.4) surpass all the results of the second-order approaches (obtained in a perfect case without noise, $p = 0$). Moreover, it is worth mentioning that whereas the cluster and the third-order frameworks alone performed similarly, their combination reported higher results for higher noise, p above 0.4.

Further results with 3 clusters are reported in Appendix F. In this case, the original second-order formulation totally fails to learn a generalizable metric and presents random results. On the other hand, GHOST manages perfect results up to the noise of 20% (p=0.2) and very good performance up to a noise of 50% (p=0.50).

We observe that the second-order method only needs 30 seconds, while the third-order with the cluster metric needs 270 seconds.

5.2. Covid-19 Dataset

Table 3 presents the results of our GHOST framework over the Covid-19 dataset introduced in [30] and compares them to the results obtained by the authors using an Ensemble approach for both feature selection and classification. We also compared our results with an approach investigating the use of deep learning-based representations to integrate the imaging information. By leveraging the 3D contouring model proposed in [30], we extracted a deep feature space of dimension 128 of the lung. The classification ensemble approach used in [30] with classical imaging features is used as a bench-

mark. In addition, we report the results of the automated machine learning tool TPOT [31], which relies on genetic algorithms to find the best machine learning pipeline, including feature selection and classification models. This experiment highlights the interest in considering graph information as it improves the results over the basic framework. Also, it not only enables us to outperform an ensemble method over several standard classifiers leveraging carefully chosen features obtained with both field expert knowledge and an ensemble data-driven framework but also the results obtained using deep learning representation-based features and a classification pipeline defined by a state-of-the-art genetic algorithm.

Framework	**Balanced Accuracy**		**Weighted Precision**		**Weighted Sensitivity**		**Weighted Specificity**	
	Training	**Test**	**Training**	**Test**	**Training**	**Test**	**Training**	**Test**
GHOST	**0.67**	**0.71**	**0.78**	**0.8**	**0.69**	**0.73**	**0.65**	**0.69**
Ensemble	0.73	0.7	0.82	0.81	0.67	0.64	0.8	0.77
Deep Features	0.71	0.68	0.8	0.79	0.72	0.72	0.71	0.64
TPOT	0.84	0.64	0.87	0.76	0.82	0.71	0.86	0.56

Table 3: Performance of the different learning frameworks over the Covid-19 dataset. The first row is the GHOST framework, the following 3 rows are the performance of the ensemble of standard classifiers on classical imaging features as defined in [30], of the ensemble of standard classifiers on deep learning representations-based imaging features and of a feature selection and classification pipeline learned by the genetic algorithm defined in [31].

6. Conclusion

This paper proposed a novel distance learning framework to leverage higher-order information, including graph patterns and cluster-based met-

rics, towards a dedicated-to-the-task metric definition. Moreover, we demonstrated the value of leveraging graph-based information for classification. In particular, we have highlighted the interest in designing a graphical representation of data to extract structural information. Also, we studied the relevance of higher-order metrics and experimented with several directions to better account for structures in the data. In the future, we aim to study other kinds of data with known graphical representations as PPI networks. We also want to study more intricate metrics to better exploit higher-order information, for instance, Mahalanobis distance or graph-density-based information.

In this study, we have proved the efficiency of our approach on a challenging Covid-19 patient stratification task and present results outperforming ensemble approaches for feature selection and classification on common classical and deep learning representations-based imaging features as well as a dedicated pipeline automatically designed by a state-of-the-art genetic algorithm.

We show that the time complexity of our method is polynomial in the number of training objects with the order as degree. This might be a limitation on very large datasets, but GHOST remains tractable on common medical datasets of several hundreds of patients as the Covid-19 one we used. In addition, the resolution through dual decomposition is highly parallelizable by design. An implementation on GPU would greatly improve the running time. As this approach is meant to be used with expert annotation, it requires manually designing meta-features using known notions of similarity between the samples. However, this constraint is well suited for

many applications, as in network science, where metrics, including higher-order metrics, are leveraged to uncover complex patterns that could be used to define GHOST meta-features [32]. In further work, we want to investigate the definition of meta-features with a graph neural network-based method to determine a metric between groups of samples as the ones detailed in [33]. For the same reason, this method requires an annotation of the samples to learn the different cluster structures. An interesting direction we would like to pursue to alleviate the need for cluster annotation is GHOST's ability to be used with annotated and non-annotated datasets. The ground truth of the datasets without annotations can be inferred periodically using any clustering method and the current learned metric. It would allow the learning process to rely on different data structures and enforce the metric learned to present properties of a specific clustering method. Finally, we would like to leverage GHOST's ability to learn from several grounds truths and/or datasets (which number is represented by the K in our formulation) towards learning to perform a consensus of different higher-order community detection methods. This direction would be particularly valuable as it would improve the robustness of community detection, for which many definitions coexist [34].

Acknowledgment

This work was partially supported by the Fondation pour la Recherche Médicale (FRM; no. DIC20161236437) and by the ARC sign'it grant: Grant SIGNIT201801286.

References

[1] L. Yu, H. Liu, Feature selection for high-dimensional data: A fast correlation-based filter solution, in: Proceedings of the 20th international conference on machine learning (ICML-03), 2003.

[2] R. Sun, E. J. Limkin, M. Vakalopoulou, L. Dercle, S. Champiat, S. R. Han, L. Verlingue, D. Brandao, A. Lancia, S. A. et al., A radiomics approach to assess tumour-infiltrating CD 8 cells and response to anti-PD-1 or anti-PD-l1 immunotherapy: an imaging biomarker, retrospective multicohort study, The Lancet Oncology (2018).

[3] E. Battistella, M. Vakalopoulou, R. Sun, T. Estienne, M. Lerousseau, S. Nikolaev, E. A. Andres, A. Carré, S. Niyoteka, C. Robert, et al., Cancer gene profiling through unsupervised discovery, arXiv (2021).

[4] E. P. Xing, A. Y. Ng, M. I. Jordan, S. Russell, Distance metric learning with application to clustering with side-information, in: NIPS, Citeseer, 2002.

[5] S. Xiang, F. Nie, C. Zhang, Learning a mahalanobis distance metric for data clustering and classification, Pattern Recognition (2008).

[6] N. Komodakis, Learning to cluster using high order graphical models with latent variables, in: 2011 International Conference on Computer Vision, IEEE, 2011.

[7] D. Coppes, P. Cermelli, A machine-learning procedure to detect network attacks, Journal of Complex Networks 11 (2023) cnad017.

[8] E. Battistella, L. Cholvy, Modelling and simulating extreme opinion diffusion, in: International Conference on Agents and Artificial Intelligence, Springer, 2018.

[9] H. Caniza, J. J. Cáceres, M. Torres, A. Paccanaro, Landis: the disease landscape explorer, European Journal of Human Genetics (2024) 1–5.

[10] V. Hovenga, J. Kalita, O. Oluwadare, Hic-gnn: A generalizable model for 3d chromosome reconstruction using graph convolutional neural networks, Computational and Structural Biotechnology Journal 21 (2023) 812–836.

[11] E. Lee, K. Chern, M. Nissen, X. Wang, I. Consortium, C. Huang, A. K. Gandhi, A. Bouchard-Côté, A. P. Weng, A. Roth, Spatialsort: a bayesian model for clustering and cell population annotation of spatial proteomics data, Bioinformatics 39 (2023) i131–i139.

[12] S. Jin, Y. Hong, L. Zeng, Y. Jiang, Y. Lin, L. Wei, Z. Yu, X. Zeng, X. Liu, A general hypergraph learning algorithm for drug multi-task predictions in micro-to-macro biomedical networks, PLOS Computational Biology 19 (2023) e1011597.

[13] N. Somu, M. R. G. Raman, K. Kirthivasan, V. S. S. Sriram, Hypergraph based feature selection technique for medical diagnosis, Journal of Medical Systems (2016).

[14] A. Antelmi, G. Cordasco, M. Polato, V. Scarano, C. Spagnuolo, D. Yang, A survey on hypergraph representation learning, ACM Computing Surveys 56 (2023) 1–38.

[15] C. Yang, R. Wang, S. Yao, T. Abdelzaher, Semi-supervised hypergraph node classification on hypergraph line expansion, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2022.

[16] A. R. Benson, D. F. Gleich, J. Leskovec, Higher-order organization of complex networks, Science (2016).

[17] R. Lambiotte, M. Rosvall, I. Scholtes, From networks to optimal higher-order models of complex systems, Nature Physics (2019).

[18] A. K. Jain, Data clustering: 50 years beyond k-means, Pattern Recognition Letters (2010).

[19] Z. Yu, Z. Kuang, J. Liu, H. Chen, J. Zhang, J. You, H.-S. Wong, G. Han, Adaptive ensembling of semi-supervised clustering solutions, IEEE Transactions on Knowledge and Data Engineering (2017).

[20] K. Wagstaff, Refining inductive bias in unsupervised learning via constraints, in: AAAI/IAAI, 2000.

[21] I. Davidson, S. Ravi, Clustering with constraints: Feasibility issues and the k-means algorithm, in: Proceedings of the 2005 SIAM international conference on data mining, SIAM, 2005.

[22] M. T. Law, R. Urtasun, R. S. Zemel, Deep spectral clustering learning, in: International conference on machine learning, PMLR, 2017.

[23] T. Finley, T. Joachims, Supervised clustering with support vector machines, in: International conference on Machine learning, 2005.

[24] E. Battistella, M. Vakalopoulou, T. Estienne, M. Lerousseau, R. Sun, C. Robert, N. Paragios, E. Deutsch, Gene expression high-dimensional clustering towards a novel, robust, clinically relevant and highly compact cancer signature, in: IWBBIO 2019, Granada, Spain, 2019.

[25] A. Fix, A. Gruber, E. Boros, R. Zabih, A graph cut algorithm for higher-order markov random fields, in: 2011 International Conference on Computer Vision, IEEE, 2011.

[26] H. Ishikawa, Transformation of general binary mrf minimization to the first-order case, IEEE transactions on pattern analysis and machine intelligence (2010).

[27] N. Komodakis, B. Xiang, N. Paragios, A framework for efficient structured max-margin learning of high-order mrf models, IEEE transactions on pattern analysis and machine intelligence (2014).

[28] N. Komodakis, N. Paragios, G. Tziritas, Mrf energy minimization and beyond via dual decomposition, IEEE transactions on pattern analysis and machine intelligence (2010).

[29] N. Jarman, E. Steur, C. Trengove, I. Y. Tyukin, C. Van Leeuwen, Self-organisation of small-world networks by adaptive rewiring in response to graph diffusion, Scientific Reports (2017).

[30] E. Battistella, G. Chassagnon, M. Vakalopoulou, S. Christodoulidis, T.-N. Hoang-Thi, S. Dangeard, E. Deutsch, F. Andre, E. Guillo, N. Halm, et al., Ai-driven quantification, staging and outcome prediction of covid-19 pneumonia, Medical Image Analysis (2021).

[31] T. T. Le, W. Fu, J. H. Moore, Scaling tree-based automated machine learning to biomedical big data with a feature set selector, Bioinformatics (2020).

[32] Q. F. Lotito, F. Musciotto, A. Montresor, F. Battiston, Higher-order motif analysis in hypergraphs, Communications Physics 5 (2022) 79.

[33] N. Jiang, B. Ning, J. Dong, A survey of gnn-based graph similarity learning, in: 2023 8th International Conference on Image, Vision and Computing (ICIVC), IEEE, 2023, pp. 650–654.

[34] A. K. Dey, Y. Tian, Y. R. Gel, Community detection in complex networks: From statistical foundations to data science applications, Wiley Interdisciplinary Reviews: Computational Statistics 14 (2022) e1566.

Appendix A. Optimizing over $\{\hat{\boldsymbol{x}}^{\boldsymbol{k,p}}\}$

For a fixed p.

$$
\begin{aligned}
&\bar{E}_p^k(x,w) = \sum_{p,q\neq p} \bar{u}_{p,q}^k(x_{p,q}) + \sum_{p,p',q\neq p} \bar{u}_{p,p',q}^k(x_{p,q}x_{p',q}) + \\
&\sum_q \bar{\phi}_{p,q}(x_{p,q}) + \bar{\phi}_p(x_p) - \frac{\beta}{|V^k|} + \\
&\sum_q (\frac{1}{|V^k|+1}(\bar{u}_{q,q}^k(x_{q,q}^k) + \sum_{p'} \bar{u}_{q,p',q}^k(x_{q,q}^k x_{p',q}^k)) + \lambda_{p,q}x_{q,q}) \\
&\qquad = \sum_{p,q\neq p} \bar{u}_{p,q}^k(1)x_{p,q} + \sum_{p,p',q\neq p} \bar{u}_{p,p',q}^k(1)x_{p,q}x_{p',q} \\
&+ \sum_q \bar{\phi}_{p,q}(x_{p,q}) + \bar{\phi}_p(x_p) - \frac{\beta}{|V^k|} + \\
&\sum_q (\frac{1}{|V^k|+1}(\bar{u}_{q,q}^k(1)x_{q,q}^k + \sum_{p'} \bar{u}_{q,p',q}^k(1)x_{q,q}^k x_{p',q}^k) \\
&+ \lambda_{p,q}x_{q,q}) \\
&\qquad = \sum_{p,q\neq p} (\bar{u}_{p,q}^k(1) + \sum_{p,p',q\neq p} \bar{u}_{p,p',q}^k(1)x_{p',q})x_{p,q} + \\
&\sum_q \bar{\phi}_{p,q}(x_{p,q}) + \bar{\phi}_p(x_p) - \frac{\beta}{|V^k|} \\
&+ \sum_q (\frac{1}{|V^k|+1}(\bar{u}_{q,q}^k(1) + \sum_{p'} \bar{u}_{q,p',q}^k(1)x_{p',q}^k) + \lambda_{p,q})x_{q,q}
\end{aligned}
\tag{A.1}
$$

We still have here a complex CRF energy to minimize. To solve this problem in general settings, we could use a replacement strategy leveraging the binary nature of the variables as proposed in [25]. However, it would lead to a costly optimization that would hinder the whole framework's tractability. Thus, we exploit the particularity of our distance learning task and impose a positivity constraint over the distance. Thus, $\forall p, p', q,\ \bar{u}_{p,p',q}^k(1) > 0$ and we have no

constraint on $x_{p',q}$, then, fixing $\forall p' \neq p, q,\ x_{p',q} = 0$ will decrease the objective function. So, it comes down to:

$$\begin{aligned}\min_x \bar{E}_p^k(x,w) = \min_x (&\sum_p \sum_{q \neq p} (\bar{u}_{p,q}^k(1) + \bar{u}_{p,q,q}^k(1) x_{q,q} + \\ &\bar{u}_{p,p,q}^k(1)) x_{p,q} + \sum_q \bar{\phi}_{p,q}(x_{p,q}) + \bar{\phi}_p(x_p) \\ &- \frac{\beta}{\mid V^k \mid} + \sum_q (\frac{1}{\mid V^k \mid +1} (\bar{u}_{q,q}^k(1) + \bar{u}_{q,q,q}^k(1)) + \lambda_{p,q}) x_{q,q})\end{aligned} \tag{A.2}$$

Minimizing $\bar{E}_p^k(x,w)$ requires the constraints $\bar{\phi}_p(x_p) = 0$ and $\bar{\phi}_{p,q}(x_{p,q}) = 0$ as the alternative is an infinite cost. It imposes that there exists one and only one q such that $x_{p,q} = 1$ and for that q, $x_{q,q} = 1$. Thus, $\forall q$, if we denote $\theta_q^k = \frac{\bar{u}_{q,q}^k(1) + \bar{u}_{q,q,q}^k(1)}{\mid V^k \mid +1} + \lambda_{p,q}$ and $\bar{\theta}_q^k = [\theta_q^k]_+ + \bar{u}_{p,q}^k(1) + \bar{u}_{p,q,q}^k(1) + \bar{u}_{p,p,q}^k(1)$ with $[z]_+ = max(0,z)$, the terms containing $x_{q,q}$ are $(u_{p,q,q} x_{p,q} + \theta_q^k) x_{q,q}$. Then, to decrease our objective function, we have to set $x_{q,q} = 1$ if $\theta_q^k < 0$ and the cost $u_{p,q,q}$ will only be paid if p is assigned to q. Regarding this assignment, the cost of $x_{p,q} = 1$ will be minimal iff $q = \arg\min \bar{\theta}_q^k$ where the term $[\theta_q^k]_+$ accounts for the extra cost of satisfying $x_{p,q} = 1 \implies x_{q,q} = 1$ will entail if $x_{q,q}$ did not verify $\theta_q^k < 0$ and so would have been set to 0.

Lemma 4.

$$\begin{aligned}\hat{x}_{q,q}^p &= [\theta_q^k < 0] \\ \hat{x}_{p,q}^p &= [q = \bar{q}] \ where\ \bar{q} = argmin_q(\bar{\theta}_q^k)\end{aligned} \tag{A.3}$$

$$with\ \theta_q^k = \frac{\bar{u}_{q,q}^k(1) + \bar{u}_{q,q,q}^k(1)}{\mid V^k \mid +1} + \lambda_{p,q}\ and\ \bar{\theta}_q^k = [\theta_q^k]_+ + \bar{u}_{p,q}^k(1) + \bar{u}_{p,q,q}^k(1) + \bar{u}_{p,p,q}^k(1)$$

Appendix B. Optimizing over $\{\hat{x}^{k,C}\}$

For a fixed C.

$$\begin{aligned}\bar{E}_C^k(x,w) =&\bar{\phi}_C(x_C) + \sum_{q\in C}(\frac{1}{\mid V^k \mid +1}(\bar{u}_{q,q}^k(x_{q,q}^k) \\ &+ \sum_{p'} \bar{u}_{q,p',q}^k(x_{q,q}^k x_{p',q})) + \lambda_C x_C)\end{aligned} \tag{B.1}$$

As previously, for better traceability, we will enforce the positivity constraint on our distance, so $\forall p' \neq q$, $x_{p',q} = 0$ as $\bar{u}_{q,p',q}^k > 0$ and we have no constraint over $x_{p',q}$ when $p' \neq q$.

Regarding $x_{q,q}$, we consider two cases:

(i) $\forall q \in C$, $\hat{x}_{q,q}^{k,C} = 0$. Then, the optimal energy is $OPT_1 = -\alpha \mid C^k \mid$.

(ii) $\exists q$, $\hat{x}_{q,q}^{k,C} = 1$. Then:

$$\begin{aligned}\bar{E}_C^k(x,w) =&\bar{\phi}_C(x_C) + \sum_{q\in C}(\frac{1}{\mid V^k \mid +1}(\bar{u}_{q,q}^k(x_{q,q}^k) + \bar{u}_{q,q,q}^k(x_{q,q}^k))+ \\ &\lambda_C x_C) \\ =&\sum_{q\in C}(\frac{1}{\mid V^k \mid +1}((\bar{u}_{q,q}^k(1) + \bar{u}_{q,q,q}^k(1))x_{q,q}^k) + \lambda_{C_q} x_C) \\ &- \alpha(\mid V^k \mid - \mid C^k \mid)(\sum_{q\in C} x_{q,q}^k - 1) \\ =&\sum_{q\in C}(\frac{1}{\mid V^k \mid +1}(\bar{u}_{q,q}^k(1) + \bar{u}_{q,q,q}^k(1)) + \lambda_{C_q} \\ &- \alpha(\mid V^k \mid - \mid C^k \mid))x_{q,q}^k + \alpha(\mid V^k \mid - \mid C^k \mid)\end{aligned}$$

In this case, $\forall q \in C$,

$$\begin{aligned}\hat{x}_{q,q} = 1 \iff \frac{1}{\mid V^k \mid +1}(\bar{u}_{q,q}^k(1) + \bar{u}_{q,q,q}^k(1) + \lambda_{C_q} \\ -\alpha(\mid V^k \mid - \mid C^k \mid) < 0\end{aligned}$$

i.e.

$$\hat{x}_{q,q} = [\frac{1}{\mid V^k \mid +1}(\bar{u}^k_{q,q}(1) + \bar{u}^k_{q,q,q}(1)) + \lambda_{C_q} < \alpha(\mid V^k \mid - \mid C^k \mid)]$$

And, in this case, the optimal energy is

$$OPT_2 = \sum_{q \in C}[\frac{1}{\mid V^k \mid +1}(\bar{u}^k_{q,q}(1) + \bar{u}^k_{q,q,q}(1)) + \lambda_{C_q} - \alpha(\mid V^k \mid - \mid C^k \mid)]_- + \alpha(\mid V^k \mid - \mid C^k \mid)$$

with $[z]_- = min(0, z)$.

Finally, the second case holds true iff

$$\begin{aligned} & OPT_2 < OPT_1 \\ & \iff \sum_{q \in C}[\frac{1}{\mid V^k \mid +1}(\bar{u}^k_{q,q}(1) + \bar{u}^k_{q,q,q}(1)) + \lambda_{C_q} \\ & - \alpha(\mid V^k \mid - \mid C^k \mid)]_- + \alpha(\mid V^k \mid - \mid C^k \mid) < -\alpha \mid C^k \mid \\ & \iff \sum_{q \in C}[\frac{1}{\mid V^k \mid +1}(\bar{u}^k_{q,q}(1) + \bar{u}^k_{q,q,q}(1)) + \lambda_{C_q} \\ & - \alpha(\mid V^k \mid - \mid C^k \mid)]_- + \alpha \mid V^k \mid < 0 \end{aligned}$$

Lemma 5. *For fixed $C \in \mathcal{C}^k$, let $\theta^k_q = \frac{1}{\mid V^k \mid +1}\bar{u}^k_{q,q}(1) + \bar{u}^k_{q,q,q}(1) + \lambda_{C_q}$, $\forall q \in C$. A minimizer $\hat{x}^C$ of $\bar{E}^k_C(x, w, \lambda^k)$ is given by*

$$\forall q \in C,\ \hat{x}^C_{q,q} = \begin{cases} [\theta^k_q < \alpha(\mid V^k \mid - \mid C^k \mid)], \\ \quad if\ \sum_{q' \in C}[\theta^k_q - \alpha(\mid V^k \mid - \mid C^k \mid)]_- + \\ \qquad \alpha \mid V^k \mid < 0 \\ 0\ otherwise \end{cases}$$

with $[z]_- = min(0, z)$

Appendix C. Optimizing over $\boldsymbol{\lambda}$ and $\boldsymbol{w}$

These updates are defined from subgradients of the objective function. We compute the partial derivatives of each subproblem's hinge loss. We denote by $\hat{x}^{k,p}$ and $\hat{x}^{k,C}$ binary minimizers of energies $\bar{E}^k_p(x,w)$ and $\bar{E}^k_C(x,w)$.

$$\begin{aligned}
\delta w^{k,p} =& \sum_{p',q\neq p} x^k_{p,q} x^k_{p',q} f^k_{p,p',q} + \sum_{q\neq p} x^k_{p,q} f^k_{p,q} + \\
& \sum_q \frac{1}{\mid V^k \mid +1} (x^k_{q,q} f^k_{q,q} + \sum_{p'} x^k_{q,q} x^k_{p',q} f^k_{qp',q}) - \\
& (\sum_{q\neq p} \hat{x}^{k,p}_{p,q} \hat{x}^{k,p}_{p',q} f^k_{p,p',q} + \sum_{q\neq p} \hat{x}^{k,p}_{p,q} f^k_{p,q} + \\
& \sum_q \frac{1}{\mid V^k \mid +1} (\hat{x}^{k,p}_{q,q} f^k_{q,q} + \sum_{p'} \hat{x}^{k,p}_{q,q} \hat{x}^{k,p}_{p',q} f^k_{qp',q})) \\
\delta w^{k,C} =& \sum_q \frac{1}{\mid V^k \mid +1} (x^k_{q,q} f^k_{q,q} + \sum_{p'} x^k_{q,q} x^k_{p',q} f^k_{qp',q}) - \\
& \sum_q \frac{1}{\mid V^k \mid +1} (\hat{x}^{k,C}_{q,q} f^k_{q,q} + \sum_{p'} \hat{x}^{k,C}_{q,q} \hat{x}^{k,C}_{p',q} f^k_{qp',q}) \\
\delta \lambda^{k,p} =& x^k_{q,q} - \hat{x}^{k,p}_{q,q} \\
\delta \lambda^{k,C} =& x^k_{q,q} - \hat{x}^{k,C}_{q,q}
\end{aligned} \tag{C.1}$$

Thus, we have the update:

$$\begin{aligned}
\delta w =& \tau \nabla J(w) + \sum_k (\sum_p \delta w^{k,p} + \sum_C \delta w^{k,C}) \\
=& \tau \nabla J(w) + \sum_k \delta^k w
\end{aligned} \tag{C.2}$$

with:

$$\begin{aligned}\delta_w^k = &\sum_{p,p',q} x_{p,q}^k x_{p',q}^k f_{p,p',q}^k + \sum_q x_{p,q}^k f_{p,q}^k - \\ &(\sum_{q\neq p} \hat{x}_{p,q}^{k,p} \hat{x}_{p',q}^{k,p} f_{p,p',q}^k + \sum_{q\neq p} \hat{x}_{p,q}^{k,p} f_{p,q}^k + \\ &\sum_q \frac{1}{\mid V^k \mid +1} (\hat{X}_q^k f_{q,q}^k + \sum_{p'} \hat{X}_{p',q}^k f_{qp',q}^k))\end{aligned} \tag{C.3}$$

with $\hat{X}_q^k = \hat{x}_{q,q}^{k,C} + \sum_p \hat{x}_{q,q}^{k,p}$ and $\hat{X}_{p',q}^k = \sum_p \hat{x}_{q,q}^{k,p} \hat{x}_{p',q}^{k,p} + \hat{x}_{q,q}^{k,C} \hat{x}_{p',q}^{k,C}$

Finally, to obtain an acceptable solution λ we need to project on set Λ. To that purpose, we simply have to substract by $\frac{\lambda^{k,C} + \sum_p \lambda^{k,p}}{\mid V^k \mid +1} = x_{q,q}^k - \frac{\hat{X}_q^k}{\mid V^k \mid +1}$. Thus, we have to update w and λ with the following formulas:

Lemma 6. *Let s_t be the weight accorded to the optimization at step t. We define $\hat{X}_q^k = \hat{x}_{q,q}^{k,C} + \sum_p \hat{x}_{q,q}^{k,p}$ and $\hat{X}_{p',q}^k = \hat{x}_{q,q}^{k,p} \hat{x}_{p',q}^{k,p} + \hat{x}_{q,q}^{k,C} \hat{x}_{p',q}^{k,C}$. Then, update reduces to:*

$$\begin{aligned} w \; -&= s_t(\tau \nabla J(w) + \sum_k \delta_w^k) \\ \lambda_{p,q}^k \; -&= s_t(\frac{\hat{X}_q^k}{\mid V^k \mid +1} - \hat{x}_{q,q}^{k,p}) \\ \lambda_{p,q}^k \; -&= s_t(\frac{\hat{X}_q^k}{\mid V^k \mid +1} - \hat{x}_{q,q}^{k,C}) \end{aligned} \tag{C.4}$$

where

$$\begin{aligned}\delta_w^k = &\sum_{p,p',q} x_{p,q}^k x_{p',q}^k f_{p,p',q}^k + \sum_{p,q} x_{p,q}^k f_{p,q}^k - \\ &(\sum_{p',q\neq p} \hat{x}_{p,q}^{k,p} \hat{x}_{p',q}^{k,p} f_{p,p',q}^k + \sum_{q\neq p} \hat{x}_{p,q}^{k,p} f_{p,q}^k + \\ &\sum_q \frac{1}{\mid V^k \mid +1} (\hat{X}_q^k f_{q,q}^k + \sum_{p'} \hat{X}_{p',q}^k f_{qp',q}^k))\end{aligned} \tag{C.5}$$

Appendix D. Generalization to Higher-Order Distances

Our approach's strength is its ability to be efficiently generalized to any order. Let $h \geq 2$ be the order of the distance we are looking for. h corresponds to the maximal set size we will consider in our metric definition. Our target metric considers any set S of size $\mid S \mid \leq h$ and is defined as $d_S = w^T f_{\{p\}_{p\in S}}(y)$ where $f_{\{p\}_{p\in S}}$ is a positive feature function providing a closeness score on set S. This metric aims to establish a characterization of the meaningfulness of samples grouped in S altogether. In this case, we consider the energy defined as:

$$\begin{aligned} E(x,d)^k = &\sum_{l\in[0,h-2]} \sum_{p,p_1,\ldots,p_l,q} u^k_{p,p_1,\ldots,p_l,q}(x_{p,q} \prod_{i\in[1,l]} x_{p_i,q}, d) + \\ &\sum_{p,q} \phi_{p,q}(x_{p,q}, x_{q,q}) + \sum_p \phi_p(x_p) + \sum_C \phi_C(x_C) - \beta|\mathcal{C}^k| \end{aligned} \tag{D.1}$$

We now consider the higher order potential of order $l < h-2$, $u_{p,p_1,\ldots,p_l,q}(\prod_{i\in[1,l]} x_{p_i,q}, d)$ focusing on establishing the cost of assigning $p, p_1, ..., p_l$ to q. The potentials definitions are:

$$\begin{aligned} &u^k_{p,p_1,\ldots,p_l,q}(x_{p,q} \prod_{i\in[1,l]} x_{p_i,q}, d) = d^k_{p,p_1,\ldots,p_l,q} x_{p,q} \prod_{i\in[1,l]} x_{p_i,q} \\ &d^k_{p,\prod_{i\in[1,l]} p_i} x_{p,q} \prod_{i\in[1,l]} x_{p_i,q} = w^T f_{p,p_1,\ldots,p_l,q}(y^k) \\ &\phi_{p,q}(x_{p,q}, x_{q,q}) = \delta(x_{p,q} \leq x_{q,q}) \\ &\phi_p(x_p) = \delta(\sum_q x_{p,q} = 1) \end{aligned} \tag{D.2}$$

First, regarding the optimization over $\{x^k\}$ for a fixed vector w. As previously, the satisfaction of the constraints induces:

$$x^k = \arg\min_{x\in\mathcal{X}(C^k)} (\sum_{l\in[0,h-2]} \sum_{p,p_1,\ldots,p_l,q} d^k_{p,p_1,\ldots,p_l,q} x_{p,q} \prod_{i\in[1,l]} x_{p_i,q}) \tag{D.3}$$

And, again, this problem's minimization only requires the set Q^k of exemplars q minimizing the above function per cluster in $\mathcal{C}^k$. Then, we assign each point of the cluster to its exemplar. In this case, the constraints will be satisfied as we ensure each cluster has one and only one center and assign all cluster samples to this center.

As before, for each $k < K$, we define a slave problem per datapoint $p \in V^k$, $\bar{E}_p^k$, and one per cluster $C \in \mathcal{C}^k$, $\bar{E}_C^k$.

$$\begin{aligned}
&\bar{E}_p^k(x, w) = \sum_{l \in [0,h-2]} \sum_{p_1,\ldots,p_l,q \neq p} u^k_{p,p_1,\ldots,p_l,q}(x_{p,q} \prod_{i \in [1,l]} x_{p_i,q}, d) \\
&+ \sum_q \bar{\phi}_{p,q}(x_{p,q}) + \bar{\phi}_p(x_p) - \frac{\beta}{\mid V^k \mid} + \\
&\sum_q (\frac{1}{\mid V^k \mid +1} (\sum_{l \in [0,h-2]} \sum_{p_1,\ldots,p_l} u^k_{q,p_1,\ldots,p_l,q}(x_{q,q} \prod_{i \in [1,l]} x_{p_i,q}, d)) + \lambda_{p,q} x_{q,q}) \qquad \text{(D.4)} \\
&\bar{E}_C^k(x, w) = \bar{\phi}_C(x_C) + \\
&\sum_q (\frac{1}{\mid V^k \mid +1} (\sum_{l \in [0,h-2]} \sum_{p_1,\ldots,p_l} u^k_{q,p_1,\ldots,p_l,q}(x_{q,q} \prod_{i \in [1,l]} x_{p_i,q}, d) + \lambda_{C_q} x_{q,q})
\end{aligned}$$

where the Lagrangian variables $\lambda = \{\{\lambda_{p,q}\}, \{\lambda_{C_q}\}\}$ are used to ensure the consistency of the solution. We impose the satisfaction of: $\lambda \in \Lambda^k = \{\lambda : \sum_{p \in S^k} \lambda_{p,q} + \lambda_{C_q} = 0, \forall C \in \mathcal{C}^k, q \in C\}$. Therefore, by design, $\bar{E}^k(x^k, w) = \sum_p \bar{E}_p^k(x, w) + \sum_C \bar{E}_C^k(x, w)$. Thus, finally, the lost function to be minimized is:

$$\min_{\{x^k \in \mathcal{X}(\mathcal{C}^k)\}, w, \{\lambda^k \in \Lambda^k\}} \tau J(w) + \sum_k \sum_{p \in V^k} \mathcal{L}_{\bar{E}_p^k} + \sum_k \sum_{C \in \mathcal{C}^k} \mathcal{L}_{\bar{E}_C^k} \qquad \text{(D.5)}$$

Appendix D.0.1. Optimizing over $\{\hat{x}^{k,p}\}$

Regarding the point-wise subproblems, we proceed as follows. In the following lemma, we generalize the solution that was proposed in Lemma 1

to a general order configuration as:

Lemma 7. *For fixed $p \in V^k$, let $\theta_q^k = \frac{\sum_{l\in[0,h-2]} u^k_{q,\dots,q}(1)}{|V|^k+1} + \lambda^k_{p,q}$ and $\bar{\theta}_q^k = [\theta_q^k]_+ + \sum_{l\in[0,h-2]} \sum_{p_1,\dots,p_l\in\{p,q\}} u^k_{p,p_1,\dots,p_l,q}$ where $[z]_+ = max(0,z)$. minimizer $\hat{x}^p$ of $\bar{E}^k_p(x,w,\lambda^k)$ is given by*

$$\hat{x}^{k,p}_{q,q} = [\theta^k_q < 0], \quad \hat{x}^{k,p}_{p,q} = [q = \bar{q}] \text{ where } \bar{q} = \arg\min_q(\bar{\theta}^k_q) \tag{D.6}$$

Appendix D.0.2. Optimizing over $\{\hat{x}^{k,C}\}$

Regarding the cluster-wise subproblems, we generalize the Lemma 2 with:

Lemma 8. *For fixed $C \in \mathcal{C}^k$, let $\theta^k_q = \frac{\sum_{l\in[0,h-2]} u^k_{q,\dots,q}(1)}{\mid V^k \mid +1} + \lambda^k_{C_q}$, $\forall q \in C$. A minimizer $\hat{x}^{k,C}$ of $\bar{E}^k_C(x,w,\lambda^k)$ is given, $\forall q \in C$, by*

$$\hat{x}^{k,C}_{q,q} = \begin{cases} [\theta^k_q < \alpha(\mid V^k \mid - |C^k|)], \\ \quad if \sum_{q'\in C}[\theta^k_q - \alpha(\mid V^k \mid - |C^k|)]_- + \alpha \mid V^k \mid < 0 \\ 0 \text{ otherwise} \end{cases} \tag{D.7}$$

with $[z]_- = min(0,z)$.

Appendix D.0.3. Optimizing over λ and w

Lemma 9. *Let s_t be the weight granted to the optimization at step t. We define $\hat{X}^k_q = \hat{x}^{k,C}_{q,q} + \sum_p \hat{x}^{k,p}_{q,q}$ and $\hat{X}^k_{p,\prod_{i\in[1,l]} p_i,q} = \hat{x}^{k,p}_{q,q} \prod_{i\in[1,l]} \hat{x}^{k,p}_{p_iq} + \hat{x}^{k,C_q}_{q,q}[p_i = q, \forall i \in [1,l]]$. Then, the updates reduce to:*

$$\begin{aligned} w \mathrel{-}= & s_t(\tau\nabla J(w) + \sum_k \delta^k_w), \quad \lambda^k_{p,q} \mathrel{-}= s_t(\frac{\hat{X}^k_q}{\mid V^k \mid +1} - \hat{x}^{k,p}_{q,q}) \\ \lambda^k_{C_q} \mathrel{-}= & s_t(\frac{\hat{X}^k_q}{\mid V^k \mid +1} - \hat{x}^{k,C}_{q,q}) \end{aligned} \tag{D.8}$$

where

$$\delta_w^k = \sum_{l\in[0,h-2]} \sum_{p,p_1,\ldots,p_l,q\in V^k} f_{p,p_1,\ldots,p_l,q}(x^k_{p,q} \prod_{i\in[1,l]} x^k_{p_i,q}, d) - \\ \sum_{q\in V^k}(\hat{X}^k_q f^k_{q,q} + \sum_{l\in[0,h-2]}(\sum_{p\neq q\in V^k} \sum_{p_i\in\{p,q\}\forall i\in[1,l]} \hat{x}^{k,p}_{p,q}\hat{x}^{k,p}_{q,q} f^k_{p,p_1,\ldots,p_l,q} + \\ \frac{1}{\mid V^k \mid +1} \sum_{p\in V^k} \sum_{p_i\in\{p,q\}\forall i\in[1,l]} \hat{X}^k_{p,\prod_{i\in[1,l]} p_i,q} f^k_{q,\prod_{i\in[1,l]} p_i,q})) \tag{D.9}$$

Appendix E. Extension to Cluster Metrics

A final interesting addition we can bring to our higher-order distance learning framework is to consider a metric between a sample and a ground truth cluster. The difference between this particular setting and the previous higher-order metrics is that here we will consider a metric able to tackle sets of objects of different sizes (the size of the clusters) and thus will not have a defined order. The interest in such a distance is to benefit from a structural metric characterizing the closeness between a sample and a given cluster. During inference, it will be especially valuable for identifying the most suited cluster for a sample.

We formulate this new problem by adding to the higher-order distance defined in the previous section a term $w^T f_{p,C}(y^k)$ for any $p \in V^k$ and any $C \in \mathcal{C}^k$. Then, from the previously defined energy $E(x,d)^k$ we will define our new energy

$$E^{k*}(x,d) = E^k(x,d) + \sum_{p\in V^k} \sum_{C\in\mathcal{C}^k} w^T f_{p,C}(y^k) \sum_{q\in C} x_{p,q}$$

where we penalize the assignment of a sample p to a center q by the distance between the sample and the center's cluster according to given cluster-wise

feature functions. First, regarding the optimization over $\{x^k\}$ for a fixed vector w. As previously, the satisfaction of the constraints induces to find the cluster centers q minimizing for its cluster C:

$$\arg\min_{q\in C}(\sum_{l\in[0,h-2]}\sum_{p_1,..,p_l,p\in C} d^k_{p,p_1,...,p_l,q}x_{p,q}\prod_{i\in[1,l]}x_{p_i,q}+\sum_{p\in C}w^T f_{p,C}(y^k)x_{p,q}) \quad \text{(E.1)}$$

x^k is inferred by assigning each sample of a cluster to the cluster center. We modify the cluster-wise slave problems of the dual decomposition as follows:

$$\bar{E}^{k*}_C(x,w)=\bar{E}^k_C(x,w)+\sum_{p\in V^k}w^T f_{p,C}(y^k)\sum_{q\in C}x_{p,q}+\frac{1}{2(\mid V^k\mid+1)}w^T f_{q,C}(y^k)\sum_{q\in C}x_{q,q} \quad \text{(E.2)}$$

with $\bar{E}^k_C(x,w)$ the energy defined as in equation D.4. Therefore, the cluster-wise slave resolution is now:

Lemma 10. *For fixed $C\in\mathcal{C}^k$. Let $\theta^{k*}_q=\frac{\sum_{l\in[0,h-2]}u^k_{q,...,q}(1)+f_{q,C}(y^k)}{\mid V^k\mid+1+\lambda^{k*}_{C_q}}$, $\forall q\in V^k$. A minimizer $\hat{x}^{k,C*}$ of $\bar{E}^k_{k,C*}(x,w,\lambda^{k*})$ is given, $\forall q\in C$, by:*

$$\hat{x}^{k,C*}_{q,q}=\begin{cases}[\theta^{k*}_q<\alpha(\mid V^k\mid-|C^k|)],\\ \quad if\ \sum_{q'\in C}[\theta^{k*}_q-\alpha(\mid V^k\mid-|C^k|)]_-+\alpha\mid V^k\mid<0\\ 0\ otherwise\end{cases} \quad \text{(E.3)}$$

The updates are defined as:

Lemma 11. *Let s_t be the weight granted to the optimization at step t. We consider $\hat{X}^k_q$ and $\hat{X}^k_{p,\prod_{i\in[1,l]}p_i,q}$ as defined in lemma 9.*

$$\begin{aligned}w^*&-=s_t(\tau\nabla J(w)+\sum_k\delta^{k*}_w),\quad \lambda^{k*}_{p,q}-=s_t(\frac{\hat{X}^k_q}{\mid V^k\mid+1}-\hat{x}^{k,p}_{q,q})\\ \lambda^{k*}_{C_q}&-=s_t(\frac{\hat{X}^k_q}{\mid V^k\mid+1}-\hat{x}^{k,C*}_{q,q})\end{aligned} \quad \text{(E.4)}$$

where

$$\delta_w^{k*} = \delta_w^k + \sum_{p \in V^k} \sum_{C \in \mathcal{C}^k} f_{p,C}(y^k) \sum_{q \in C} x_{p,q}^{k*} - \sum_{p \in V^k} \sum_{C \in \mathcal{C}^k} f_{p,C}(y^k) \sum_{q \in C} (x_{p,q}^{k,C*} + x_{p,q}^{k,p}) + \left(\frac{\hat{X}_q^k}{\mid V^k \mid + 1} \sum_{q \neq p \in C} (x_{p,q}^{k,C*} + x_{p,q}^{k,p})\right) \tag{E.5}$$

Appendix F. Assessment metrics

Regarding the classification metrics, we relied on criteria allowing multi-class classification with unbalanced classes as Balanced Accuracy (BA), Weighted Precision (WP), Weighted Recall (WR), and Weighted F1-score (WF). In a general multi-class setting for a set of classes C, we consider each class $c \in C$ in one-versus-rest. We denote the set of true positives of class c as y_c, and the set of samples predicted positive for class c as y_c^P. We define the precision (P), recall (R), and F1-score (F) of a class $c \in C$ against all the other classes as:

$$P(c) = \frac{\mid y_c^P \cap y_c \mid}{\mid y_c^P \mid}, \; R(c) = \frac{\mid y_c^P \cap y_c \mid}{\mid y_c \mid}, \; F(c) = \frac{2P(c)R(C)}{2P(c) + R(c)}$$

Recall corresponds to the True Positive Rate for the class c. Precision is the Positive Predictive Value. F1-score represents the harmonic mean between R and P. Then, the weighted version WS of a metric $S \in \{P, R, F\}$ corresponds to the average of S over all classes of C weighted by the size of the classes:

$$WS = \frac{1}{\sum_{c \in C} \mid y_c \mid} \sum_{c \in C} \mid y_c \mid S(c)$$

Finally, the balanced accuracy is the proportion of correctly predicted samples normalized by the number of samples in the class. Formally, it is defined

as:

$$BA = \frac{1}{\sum_{c \in C} \mid y_c \mid} \sum_{c \in C} \frac{\mid y_c^P \cap y_c \mid}{\mid y_c \mid}$$

Appendix G. Simulated Benchmark with 3 Clusters

Second-order Baseline	Balanced Accuracy			Weighted Precision			Weighted Sensitivity			Weighted Specificity		
	Training	**Validation**	**Test**	**Training**	**Validation**	**Test**	**Training**	**Validation**	**Test**	**Training**	**Validation**	**Test**
Baseline	1	0.33	0.33	1	0.07	0.06	1	0.25	0.25	1	0.75	0.75
GHOST	**Balanced Accuracy**			**Weighted Precision**			**Weighted Sensitivity**			**Weighted Specificity**		
p	**Training**	**Validation**	**Test**	**Training**	**Validation**	**Test**	**Training**	**Validation**	**Test**	**Training**	**Validation**	**Test**
0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
0.1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
0.2	1	0.92	1	1	0.93	1	1	0.9	1	1	0.96	1
0.3	1	0.66	0.92	1	0.43	0.93	1	0.6	0.89	1	0.86	0.96
0.4	0.97	0.55	0..72	0.97	0.81	0.85	0.97	0.47	0.67	0.99	0.82	0.89
0.5	0.73	0.34	0.33	0.87	0.21	0.64	0.73	0.26	0.26	0.91	0.75	0.75

Appendix G.1. Convergence Criterion Evolution

Here, we report a Figure of the convergence criterion evolution through the different training steps for GHOST and the second-order formulation. While the values are not comparable, it allows us to compare the convergence time for each algorithm and the number of steps required in the challenging case of distance learning with three clusters.

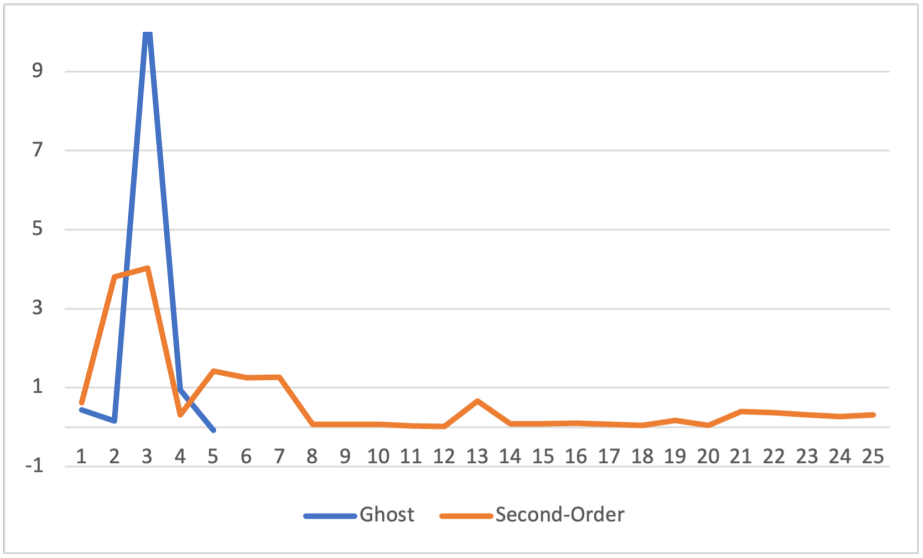


Figure G.2: **Convergence Criterion Evolution through the learning process.**