A graph-structured distance for heterogeneous datasets with meta variables [*][†]

Edward Hallé-Hannan [‡] Charles Audet [§] Youssef Diouane [¶]
Sébastien Le Digabel [‖] Paul Saves [**]

May 20, 2024

Abstract

Heterogeneous datasets emerge in various machine learning or optimization applications that feature different data sources, various data types and complex relationships between variables. In practice, heterogeneous datasets are often partitioned into smaller well-behaved ones that are easier to process. However, some applications involve expensive-to-generate or limited size datasets, which motivates methods based on the whole dataset. The first main contribution of this work is a modeling graph-structured framework that generalizes state-of-the-art hierarchical, tree-structured, or variable-size frameworks. This framework models domains that involve heterogeneous datasets in which variables may be continuous, integer, or categorical, with some identified as meta if their values determine the inclusion/exclusion or affect the bounds of other so-called decreed variables. Excluded variables are introduced to manage variables that are either included or excluded depending on the given points. The second main contribution is the graph-structured distance that compares extended points with any combination of included and excluded variables: any pair of points can be compared, allowing to work directly in heterogeneous datasets with meta variables. The contributions are illustrated with some regression experiments, in which the performance of a multilayer perceptron with respect to its hyperparameters is modeled with inverse distance weighting and K-nearest neighbors models.

Keywords. Machine learning, numerical optimization, heterogeneous datasets, distances, meta variables.

AMS subject classifications. 62J02, 62R07, 68P01, 68T09, 90C30

[*] GERAD and Department of Mathematics and Industrial Engineering, Polytechnique Montréal.

[†] DTIS, ONERA and Fédération ENAC ISAE-SUPAERO ONERA, Université de Toulouse, France.

[‡] edward.halle-hannan@polymtl.ca

[§] https://www.gerad.ca/Charles.Audet

[¶] https://www.polymtl.ca/expertises/diouane-youssef

[‖] https://www.gerad.ca/Sebastien.Le.Digabel

[**] paul.saves@onera.fr

Funding: this research was funded by a Natural Sciences and Engineering Research Council of Canada (NSERC) PhD Excellence Scholarship (PGS D), a Fonds de Recherche du Québec (FRQNT) PhD Excellence Scholarship and an Institut de l'énergie Trottier (IET) PhD Excellence Scholarship, as well as by the NSERC discovery grants RGPIN-2020-04448 (Audet), RGPIN-2024-05093 (Diouane) and RGPIN-2018-05286 (Le Digabel). The work of Saves is part of the activities of ONERA - ISAE - ENAC joint research group, and has received funding from the European Union Horizon Program under grant agreement n° 101097120.

1 Introduction

Problems that deal with heterogeneous datasets face inherent challenges that arise from several reasons, such as the generation of data from different sources, the presence of various types of variables , and data with different groups of variables. In heterogeneous datasets, data may not necessarily share the same variables, and these variables may be of various types, *i.e.*, *continuous*, *integer* and *categorical*. In practice, such datasets are often partitioned into multiple subsets of homogeneous data of similar types that are easier to tackle: this approach is undesirable when the amount of data is limited or *expensive-to-generate* [3, 21, 26]. The main motivation of this work is to exploit all accessible heterogeneous data in order to improve the performance of machine learning and data-driven optimization methods. This paper proposes a modeling framework that simultaneously considers all heterogeneous data and a distance function that compares mixed-variable data that do not necessarily share the same variables.

1.1 Scope of the work

In the present work, data heterogeneity is intrinsic to the problems addressed. More precisely, a dataset is generated from points of a domain $\mathcal{X}$ that has two characteristics that implies heterogeneity: 1) it is mixed-variable, *i.e.*, a point $x \in \mathcal{X}$ is composed of finitely many variables from any type amongst categorical (cat), integer (int) or continuous (con), and 2) two points $x, y \in \mathcal{X}$ do not necessarily share the same variables and/or are not necessarily subject to same bounds. A *point* $x \in \mathcal{X}$ is arbitrary, whereas a *data point* $x_{(i)} \in \mathcal{X}$ is a point that is known and part of a dataset. The form of the dataset depends on the sort of problem, but it is always related to a domain $\mathcal{X}$. In supervised learning or optimization, a target function $f : \mathcal{X} \to \mathbb{R}$ is addressed with a dataset $\{(x_{(i)}, f(x_{(i)}))\}_{i=1}^{N}$ of $N \in \mathbb{N}$ data couples. In unsupervised learning or clustering, the dataset $\{x_{(i)}\}_{i=1}^{M}$ consists of $M \in \mathbb{N}$ data points. This works primarily studies domains rather than datasets. The characteristics, properties and forms of datasets are seen as byproducts of the domains from which they are generated. Data fusion techniques that integrate multiple data subsets into a single heterogeneous dataset [14, 39] are not covered.

The particularity that two points in a domain $\mathcal{X}$ do not share the same variables, dimension or bounds is a consequence of the so-called *meta* variables [7]. These special variables determine if other variable(s), called *decreed*, have to be excluded or included in a point of the domain $\mathcal{X}$, or determine their admissible values. In addition to their variable type, each variable is assigned a role, such as meta or decreed, that reflects either how it influences the dimension, structure or bounds of the domain $\mathcal{X}$, or how it is subject to the influence of other variable(s). Variables that neither influence nor are influenced by other variables are assigned the *neutral* role, and they are always included in a point. The notion of roles of variables is taken from the modeling framework introduced in [7]. These roles are further developed in the present work, notably meta-decreed variables are introduced. These variables are meta variables whose own inclusions or admissible values are determined by other meta variables. The roles of variable may be illustrated on a machine learning example in which the hyperparameters of a multilayer perceptron (MLP) must be chosen, as shown by the graph structure from Figure 1.

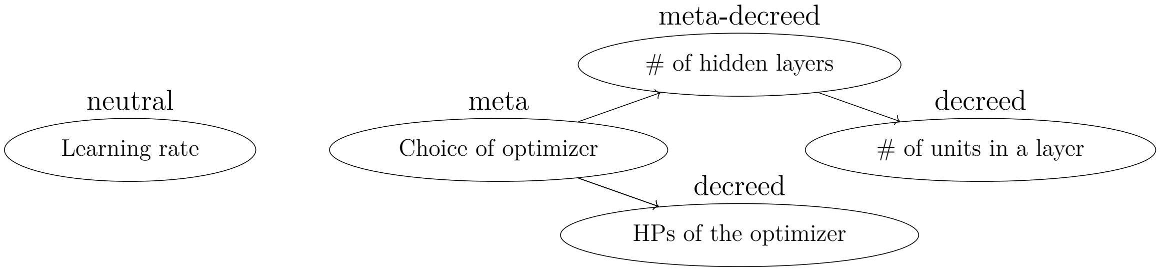

Figure 1: Graph structure for visualizing the roles of variables (hyperparameters) and the influence between hyperparameters (HPs) in the MLP example.

In the MLP example, the choice of the optimizer is a meta variable that determines the number of hidden layers that can be selected. Moreover, each optimizer has decreed variables associated to it, *e.g.*, `ADAM` is specifically associated with the running average hyperparameters. The optimizer is not a meta-decreed variable, since neither its inclusion nor its admissible values are determined by other variables. The number of hidden layers is a meta-decreed variable since its admissible values are determined by the optimizer and its value determines how many variables are present to characterize the units in the hidden layers. The variables characterizing the numbers of units are decreed but not meta as their values do not influence any other variables. The learning rate is a neutral variable since it is present in all points, and does not influence nor it is not influenced by other variables. Note that the inclusion or the admissible values of a meta-decreed variable can be determined by another meta-decreed variable: there may be several instances of meta-decreed variables. Moreover, the roles meta, meta-decreed, decreed and neutral must not be confused with the variable types, since each variable has both a type and a role.

In the MLP example, data heterogeneity emerges from two sources: 1) the diversity of variables types, *e.g.*, the optimizer that is categorical, whereas the learning rate is continuous; and 2) the meta and meta-decreed variables that determines which variables are included in point $x \in \mathcal{X}$, and the admissible values of other variable(s). The example is further detailed in Section 2.

1.2 Objectives and organization of the work

The overall objective of this work is to develop a distance function for mixed-variable domains in which two points do not necessarily share the dimension, bounds or variables. To achieve this objective, two steps are distinguished. The first step formalizes a modeling framework that thoroughly models mixed-variable domains with meta, meta-decreed, decreed and neutral variables. The framework is based on a graph structure that encompasses all information regarding the roles of variables. In [7], meta-decreed variables are prohibited. The present work generalizes the roles of variables. In the literature, some variants of graph-structured domains are referred to as tree-structured [12], hierarchical [20] or variable-size [28]: the modeling framework generalizes and unifies all these variants.

The second step constructs a mixed-variable distance function based on the modeling framework. The distance, said graph-structured, is defined on the *extended domain* $\overline{\mathcal{X}}$, rather than directly on its corresponding domain $\mathcal{X}$. The extended domain is an extension of the domain that involves all included and excluded variables. Excluded variables are those that are excluded for a given point $x \in \mathcal{X}$, but present in another point $y \in \mathcal{X}$. For example, in Figure 1, the number of units in the second hidden layer is excluded, when there is only one hidden layer. Excluded variables are taken into account by the distance, since they simultaneously provide valuable information and facilitate the comparison of two points that do not share the same variables.

The rest of the document is organized as follows. First, the remainder of this section discusses related work in Section 1.3. Then, the MLP example is further developed and detailed in Section 2. Next, the extended point $\overline{x}$ and the extended domain $\overline{\mathcal{X}}$ are thoroughly defined via the roles of variables and graph theory in Section 3. Afterwards, the graph-structured distance is defined on the extended domain $\overline{\mathcal{X}}$, which induces a distance on the original domain $\mathcal{X}$, in Section 4. Finally, computational experiments on the MLP example are carried out in Section 5 to compare the performance of two approaches on simple regression models. The first approach separates the regression problem into subproblems, each with homogeneous data, and the second one is based on the induced distance on the entire heterogeneous dataset.

1.3 Related work

Most literature on distances or similarity measures for heterogeneous datasets treats the simpler case where heterogeneity comes strictly from the variety of variable types. In classification, variants of K-nearest neighbors, based on distances (or similarity measures) that are built with combinations of continuous, integer or categorical distances, are commonly studied [1, 4, 29]. Decision trees or random forests are also utilized for classification [35], and even regression [22]. In regression, many kernel functions (similarity measures) have been recently developed for constructing Gaussian Processes (GPs) [31] over heterogeneous datasets with mixed-variable. Kernel methods are well adapted for mixed-variable problems, since mixed kernels can be directly constructed with products or additions of well-documented continuous [31], integer [17] or categorical kernels [27, 30, 33, 40].

An important reference for this work is the technical report [20], which proposes a mixed kernel function for hierarchical spaces, each paired with a directed acyclic graph (DAG), where the nodes are the variables. Variables with child nodes are required to be categorical. The kernel is constructed from one-dimensional kernels for which pseudodistances takes into account whether the variables are included or excluded. The inclusion of a variable is managed by a designated Kronecker delta function that takes the values its ancestors. In [16], a novel similarity measure, called the earth mover's intersection, computes similarity measures between sets of different sizes, similarly as the Jaccard index, but in a more technical fashion based on the earth mover's distance. In [28], GPs are constructed on said variable-design spaces, which contain dimensional variables [25] that are essentially strictly discrete meta variables that determines the inclusion or exclusion of other variables. In [11], feature models manage and capture heterogeneity across data points through tree-structured models consisting of features nodes and relationships arcs, that represent parent–child dependencies or integrity constraints [5]. Recent advances in features models addressed complex dependencies and constraints in large-scale heterogeneous datasets [9] with semantic logic.

This research is motivated by real-life machine learning and optimization applications. In deep learning, hyperparameter optimization is a highly studied problem in which the performance of a neural network is optimized with respect to its hyperparameters [15, 37, 38]. As mentioned previously, these mixed-variable problems includes the meta variable for the number of hidden layers that determines how many variables are present to characterize the units in the hidden layers. In [24], the optimization of a magnetic resonance device contains a variable that determines the number of magnets, and each additional heat intercept involves new design variables. In [13], an architecture design of an aircraft engine is optimized from a surrogate model constructed from a heterogeneous dataset, in which part of the data includes a fan and another part does not. More applications from various fields are also covered, including software architecture design [2], statistical medical research [19], drug discovery in heterogeneous datasets [36], multiple vehicle routing problem [18], and many more.

2 Illustration on the MLP example: domain of the hyperparameters

This section describes a working example inspired from [7] that models the domain of the hyperparameters of an MLP. The example is used to facilitate understanding of the contributions of this work and to perform some computational experiments. The performance of a deep model in function of its hyperparameters can be viewed as a mixed-variable function with meta variables: let $f : \mathcal{X} \to \mathbb{R}$ be a function that outputs a performance score $f(x) \in \mathbb{R}$ for a given set of hyperparameters $x \in \mathcal{X}$, where $\mathcal{X}$ is the domain of the hyperparameters. In practice, the performance score $f(x)$ is typically a score of accuracy on a untested data set, which is expensive-to-evaluate since the training, validation and performance test is done for a given and fixed set of hyperparameters [23].

The hyperparameters of the MLP example are presentend in Figure 2. In order to restrict the number of variables, some important hyperparameters are intentionally discarded, such as the momentum or the batch size. The figure is composed of several tables, which are separated into different cases of optimizer variable values. The bounds of the hyperparameters α_1, α_2, α_3, β_1, β_2 and β_3 are normalized for simplicity.

The choice of optimizer is important. First, depending on the optimizer $o \in \{\texttt{ASGD}, \texttt{ADAM}\}$, different hyperparameters are included in a point. For example, the decay α_1 is only included if $o = \texttt{ASGD}$. Second, the optimizer affects the architecture. Indeed, it influences simulta-

HP	Variable	Bounds
Learning rate	r	$]0, 1[$
Activation function	a	$\{\texttt{ReLU}, \texttt{Sig}, \texttt{Tanh}\}$
Optimizer	o	$\{\texttt{ASGD}, \texttt{ADAM}\}$

if $o = \texttt{ASGD}$

HP	Variable	Bounds
Decay	α_1	$]0, 1[$
Power update	α_2	$]0, 1[$
Average start	α_3	$]0, 1[$
# of hidden layers	l	$L_{\texttt{ASGD}}$
# of units hidden layer i	u_i	$U_{\texttt{ASGD}}$

if $o = \texttt{ADAM}$

HP	Variable	Bounds
Running average 1	β_1	$]0, 1[$
Running average 2	β_2	$]0, 1[$
Numerical stability	β_3	$]0, 1[$
# of hidden layers	l	$L_{\texttt{ADAM}}$
# of units hidden layer i	u_i	$U_{\texttt{ADAM}}$

HP	Variable	Bounds
Dropout	ρ	$\left[0, \quad \frac{\sum_{i=1}^{l} u_i}{2\tau_{\max}}\right]$

Figure 2: Hyperparameters for the MLP example.

neously the bounds of the number of hidden layers L_o, and the bounds U_o of every number of units $u_i \in U_o$, where $1 \leq i \leq l$. The subscript i in u_i represents the i-th hidden layer. The number of hidden layers $l \in L_o$ is influenced by the choice of the optimizer, and, most importantly, it determines the number of variables associated to the units. For example, if $l = 3$, then there are three hyperparameters u_1, u_2, u_3 for the units.

The number of hidden layers l and the number of units u_i influences the bounds of the dropout ρ, a regularization mechanism, via the following bounds dependencies

$$\rho \in \left[0, \quad \frac{\sum_{i=1}^{l} u_i}{2\tau_{\max}}\right] \subseteq [0, 0.5], \tag{1}$$

where $\tau_{\max}$ is a constant that consists of the maximum sum of units that can be obtained with the hyperparameters of the MLP in Figure 2, such that

$$\tau_{\max} := \max \left\{ \sum_{i=1}^{l} u_i \ : \ o \in \{\texttt{ASGD}, \texttt{ADAM}\}, \ l \in L_o, \ u_i \in U_o \text{ for } 1 \leq i \leq l \right\}. \tag{2}$$

The bounds dependencies of the dropout ρ ensure that more regularization is applied to larger architectures with more units

3 Graph-structured domains

In this section, graph-structured domains that generate heterogeneous datasets are formalized. In Section 3.1, the roles of variables are explicitly introduced. Then, excluded variables and extended point, containing all variables whether they are excluded or not, are defined in Section 3.2. In Section 3.3, some notions of graph theory are adapted for this work. Afterwards, the restricted sets, in which variables of the extended point belong, are detailed in Section 3.4. Subsequently, the extended domain $\overline{\mathcal{X}}$ is introduced in Section 3.5. Finally, Section 3.6 models the MLP example with the content introduced in Sections 3.1 to 3.5.

3.1 Roles of variables

The roles of variables are established from the the decree property that is generalized from [7]. The following definition allows a variable to simultaneously have the decree property, and have its inclusion or admissible values determined by a decree dependency (by other variables with the decree property): this is not allowed in [7].

Definition 1 (Decree property and decree dependency)**.** *The decree property is attributed to variables whose values determine if other variables are included or excluded from a point $x \in \mathcal{X}$, or whose values determine the admissible values (or bounds) of other variables.*

A decree dependency refers to the inclusion or admissible values dependency of a variable with respect to an another variable with the decree property. Variables can have multiple decree dependencies with different variables.

In Section 3.3, decree dependencies are viewed as parent-children dependencies, where the values of a parent variable determines the inclusion or admissible values of its children variables. In Figure 2, the optimizer o has the decree property, since it determines the inclusion of, among others, the decay α_1. The decay α_1 has a decree dependency with the optimizer o (parent), that is, the decay α_1 is included when $o = \texttt{ASGD}$, and excluded otherwise. In Section 3.4, the admissible values of a variable are determined by respecting its decree dependencies for given values of its parent variables. The number of units in the hidden layers also have the decree property, since they influence the bounds of the dropout ρ as presented in (1). The dropout ρ is always included, but its bounds are influenced by the values taken by the numbers of units and the number of hidden layers l. The dropout ρ has multiple decree dependencies, one with each number of units u_i and one with the number of hidden layers l.

Definition 1 on the decree property and decree dependency establishes four possible cases, which are formalized as the roles of variables in the following definition.

Definition 2 (Roles of variables)**.** *The role of a variable represents its relation to the decree property. A variable is assigned one of the following roles:*

1. *meta* (m)*, if it has the decree property, and has no decree dependency;*
2. *meta-decreed* (md)*, if it has the decree property, and has at least one decree dependency;*
3. *decreed* (dec)*, if it does not have the decree property, but has at least one decree dependency;*
4. *neutral* (neu)*, if it does not have the decree property nor decree dependency.*

Recall that the role of a variable must not be confused with its variable type. Each variable has its own variable type and is assigned its own role. In the MLP example, the optimizer $o \in \{\texttt{ADAM}, \texttt{ASGD}\}$ is a categorical variable that is assigned the meta role, since it determines the inclusion and the admissible values of other variables (decree propriety), and neither its inclusion nor its admissible values are determined by other variables (no decree dependency). For convenience, a variable and its role are referred similarly as a variable and its type, *e.g.*, the optimizer is referred as a meta categorical variable. The number of hidden layers $l \in L_o$ is a meta-decreed integer variable, since its admissible values are determined by the optimizer o (decree dependency), and it determines the inclusion of the number of units $u_1, u_2, \dots, u_l$ (decree property). The decay α_1 is a decreed continuous variable, since it does not have the decree property, and its inclusion is determined by the optimizer o. Finally, the activation function $a \in \{\texttt{ReLU}, \texttt{Sig}, \texttt{Tanh}\}$ is a neutral categorical variable, as it does not have the decree property, and it has no decree dependency.

3.2 Excluded variables and extended point

In [7], a point contains only variables that are included for the given values of variables with the decree property. In this work, variables that are excluded are also considered, since 1) it provides useful information for computing distances between two points of the domain $\mathcal{X}$ that do not share the same variables, and 2) it facilitates the computations themselves. This last remark leads to the following definition.

Definition 3 (Excluded variable)**.** *An excluded variable is a meta-decreed or decreed variable, that, for the given values of the variables associated to its decree dependencies, is not included in the given point $x \in \mathcal{X}$, but is included in at least one other point $y \in \mathcal{X}$. An excluded variable is assigned the special value* EXC *and its variable type is conserved.*

In the MLP example, the decay α_1 is a decreed continuous variable since, it is excluded when the optimizer $o = \texttt{ADAM}$, whereas $\alpha_1 \in \,]0, 1[$ (included) when $o = \texttt{ASGD}$. Definition 3 is introduced to allow the graph-structured distance to account every variable that is included in at least one point of the domain, collectively referred as all the included and excluded variables. The definition of an extended point formalizes the previous sentence.

Definition 4 (An extended point)**.** *An extended point $\overline{x}$ contains all the included and excluded variables of a corresponding point $x \in \mathcal{X}$. For $r \in R := \{\mathrm{m}, \mathrm{md}, \mathrm{dec}, \mathrm{neu}\}$ and $i \in I^r := \{1, 2, \dots, n^r\}$, the i-th variable assigned to the role r is noted $\overline{x}_i^r$, where $n^r \in \mathbb{N}$ is the number of variables assigned to the role r.*

The rest of this section discusses subtleties of Definition 4. First, the bar notation is inspired by the extended set of real numbers $\overline{\mathbb{R}} = \mathbb{R} \cup \pm\{\infty\}$, referred to as the *extended real-valued* set. The notation outlines that an extended point $\overline{x}$ is conceptually extended to incorporate excluded variables. A point $x \in \mathcal{X}$ contains only included variables, but its variables can be attributed roles, similarly as an extended point $\overline{x}$.

Second, meta and neutral variables are always included, hence there is no distinction between these variables whether they are part of an extended point $\overline{x}$ or of a point x, *i.e.*, $\overline{x}_i^r = x_i^r$ for $r \in \{\mathrm{m}, \mathrm{neu}\}$ and $i \in I^r$. The admissible values of meta and neutral variables are also fixed. In contrast, meta-decreed $\overline{x}_i^{\mathrm{md}}$ and decreed $\overline{x}_j^{\mathrm{dec}}$ variables of an extended point $\overline{x}$ may be excluded from a point x, and/or their admissible values may differ between points.

Third, the framework in [7] with meta variables but without meta-decreed variables, is a special case of the framework developed in this work. The special case can be recovered easily by removing the meta-decreed variables: this is convenient as the special case provides a specialized framework for problems of great interests, such as most hyperparameter optimization problems.

Fourth, decreed variables can be seen as bounds-dependent variables and, in that case, the bounds of an excluded variable are restricted to the empty set.

Finally, both the roles and types of a variables carry important information for properly computing distances between variables in Section 4. To avoid a cumbersome notation, variable types are not explicit, but they are implicitly considered in the computation of distances in Section 4.

Some graph theory is introduced in next Section 3.3, for further developing the variables of an extended point and their sets in Section 3.4. The extended domain $\overline{\mathcal{X}}$ is defined in Section 3.5, which allows the presentation of the graph-structured distance $\overline{\mathrm{dist}}_p : \overline{\mathcal{X}} \times \overline{\mathcal{X}} \to \overline{\mathbb{R}}^+$ in Section 4.

3.3 Notions from graph theory

At this stage of the work, Definition 1 allows meta-decreed variables to impact each others through decree dependencies. Consequently, decree dependencies between meta-decreed variables can lead to circular reasoning or contradiction. Indeed, this can be shown with a simple example with only two meta-decreed binary variables:

$$x_1^{\mathrm{md}} = \begin{cases} 0 & \text{if } x_2^{\mathrm{md}} = 1, \\ 1 & \text{if } x_2^{\mathrm{md}} = 0, \end{cases} \qquad x_2^{\mathrm{md}} = \begin{cases} 0 & \text{if } x_1^{\mathrm{md}} = 0, \\ 1 & \text{if } x_1^{\mathrm{md}} = 1, \end{cases}$$

where $x_1^{\mathrm{md}} = 0 \Rightarrow x_2^{\mathrm{md}} = 0 \Rightarrow x_1^{\mathrm{md}} = 1 \neq 0$ (contradiction). To avoid these problematic, and uncommon situations, an assumption about the decree dependencies must be introduced. Beforehand, the role graph, which among other things allows to formulate the assumption, is defined below.

Definition 5 (Role graph)**.** *The role graph $G = (V, A)$ is a graph structure, where*

- *V is the set of variables that contains all the included and excluded variables, represented as nodes,*
- *A is the set of decree dependencies that contains references for all inclusion-exclusion and admissible values dependencies between all the included and excluded variables, represented as arcs.*

An arc $a \in A$, which refers to a decree dependency, connects a parent (variable) to a child (variable), whose inclusion or admissible values are influenced by the parent. A parent is either a meta or meta-decreed $\overline{x}_i^r$, and a child is either meta-decreed or decreed, such that $a = (\overline{x}_i^r, \overline{x}_j^{r'})$, where $r \in \{\mathrm{m}, \mathrm{md}\}$ and $r' \in \{\mathrm{md}, \mathrm{dec}\}$, with $i \neq j$ when $r = r' = \mathrm{md}$. A child can have multiple parents, and vice versa.

Now that the role graph G is properly defined, the assumption that discards situations with circular reasoning or contradiction is introduced.

Assumption 1. *The role graph G is a directed acyclic graph (DAG).*

Assumption 1 ensures that the nodes in the role graph G are (partially) ordered. Circular decree dependencies, as presented in the example with two meta-decreed variables, are forbidden. To determine such a partial order, it suffices to apply a topological sort on the role graph G.

Under Assumption 1, the role graph G is a data structure that: 1) contains all the included and excluded variables in the set of variables V, 2) contains references for all inclusion-exclusion or admissible values dependencies in the set of decree dependencies A, and 3) establishes the roles of variables according to the positions of nodes in the DAG structure:

- a meta variable $\overline{x}_i^{\mathrm{m}}$ is a root node;
- a meta-decreed variable $\overline{x}_i^{\mathrm{md}}$ is an internal node with at least one parent and one child;
- a decreed variable $\overline{x}_i^{\mathrm{dec}}$ is a leaf node with at least one parent and no child;
- a neutral variable $\overline{x}_i^{\mathrm{neu}}$ is an isolated node.

Recall that meta-decreed and decreed variables are child variables whose inclusions or admissible values are determined by the values of the parents that compose their decree dependencies. The role graph G outlines that the inclusion and/or admissible values of a variable can be determined by multiple different decree dependencies, that is, from multiple parents. Hence, to determine the inclusion and/or the admissible values of a variable, it is necessary to consider simultaneously all the values of its parents. In the working example, the number of units $\overline{u}_i$ must consider simultaneously the values of its parents the optimizer $\overline{o}$, for its admissible values, and the number of hidden layers $\overline{l}$, for its inclusion. The following definition introduces formally the notion of the parents in our context.

Definition 6 (Parents)**.** *For a given role $r \in R$ and component $i \in I^r$, the parents $\overline{\mathrm{par}}_i^r$ of the variable $\overline{x}_i^r$ is the subset of variables for which there exists an arc from those variables to $\overline{x}_i^r$, i.e.,*

$$\overline{\mathrm{par}}_i^r := \{\overline{v} \in V \ : \ (\overline{v}, \overline{x}_i^r) \in A\}. \tag{3}$$

The parents are used to handle the inclusion-exclusion or the admissible values of meta-decreed and decreed variables in the next section. Note that although meta and neutral variables have no parents, they are still defined (as empty sets) for these roles in order to propose a concise expression for the extended domain $\mathcal{X}$, as in Section 3.5.

For a given variable, the inclusion or admissible values of its parents can be determined by their own parents (*i.e.*, grandparents). In Section 3.4, the ancestors of a variable, representing possible multiple generations of parents and grandparents, will be used to determine all the values that such variable can take across all possible extended points. In the MLP example, the bounds of the dropout ρ, as expressed in (1), justifies the need for ancestors. In fact, the admissible values (the bounds) of the dropout ρ are determined by the given values of its parents the number of hidden layers l and the number of units u_i. However, the constant $\tau_{\max}$ in (2), which influences all the possible values of the dropout ρ, is determined by also considering the optimizer o (grandparent). Note that the bounds of the dropout ρ do not have an explicit dependency with the values of the optimizer o since, for a given MLP problem, the constant $\tau_{\max}$ is fixed. The ancestors can be defined recursively by starting at the parents, then passing by the parents of the parents, and so on, until the roots are reached.

Definition 7 (Ancestors). *For $r \in R$ and $i \in I^r$, the ancestors of the variable $\overline{x}_i^r$, noted $\overline{\text{anc}}_i^r$, is the subset of variables that are either parents or recursively ancestors of parents of $\overline{x}_i^r$, i.e.,*

$$\overline{\text{anc}}_i^r := \overline{\text{par}}_i^r \, \cup \left(\bigcup_{\overline{x}_j^{r'} \in \, \overline{\text{par}}_i^r} \overline{\text{anc}}_j^{r'} \right) \tag{4}$$

where $\overline{\text{anc}}_j^{r'}$ denotes the ancestors of the parent variable $\overline{x}_j^{r'}$. The recursion in (4) stops at the root nodes, i.e., with $\overline{\text{par}}_j^{\text{m}} = \emptyset, \forall j \in I^{\text{m}}$.

Now that some notions of graph theory have been adapted to this work, the definition of a graph-structured domain is formally established.

Definition 8 (Graph-structured domain). *A graph-structured domain is a domain with at least one variable with the decree property, i.e., at least one meta variable.*

Definition 8 is one specific approach to formalize a graph-structured domain, yet there exist several equivalent statements.

Theorem 1 (Graph-structured domain equivalences). *Let $\mathcal{X} \neq \emptyset$, with $G = (V, A)$ as its corresponding role graph. Then the following statements are equivalent:*

1. *$\mathcal{X}$ is a graph-structured domain.*
2. *The set of decree dependencies is non-empty, i.e. $A \neq \emptyset$.*
3. *There is a least one point containing a variable with a least one parent.*

Proof. The theorem results are a direct consequence of Definitions 5 and 7. □

Theorem 1 emphasizes that the different bounds or inclusion-exclusion of variables within a graph-structured domain $\mathcal{X}$ is a consequence of interrelationships between variables, that is, its decree dependencies (see Definition 1). Note that a variable with missing entries can be modeled as a decreed variable whose inclusion is determined by an additional binary meta variable.

3.4 Universal sets and restricted sets

As mentioned in Section 3.3, meta-decreed and decreed variables are subjected to the values of their parents, since they must respect their decree dependencies for the given values of their parents. In this section, the dependencies of a variable with respect to its parents are modeled through its restricted set, that is obtained by conditioning its universal set with the values of its parents. The universal set is defined next.

Definition 9 (Universal set)**.** *For $r \in R$ and $i \in I^r$, the universal set $\overline{\mathcal{X}}_i^r$ of the variable $\overline{x}_i^r$ is the set that contains all possible values that the variable can take by considering all possible values assigned to its ancestors $\overline{\text{anc}}_i^r$.*

In the MLP example, the universal set of the number of units $\overline{u}_i$ is $\overline{U} = U_{\texttt{ASGD}} \cup U_{\texttt{ADAM}} \cup \{\texttt{EXC}\}$. The number of units $\overline{u}_i$ can be excluded, depending on the number of hidden layers $\overline{l}$, hence its universal set must contain the special value EXC.

As discussed in the previous section, the values of the ancestors must be considered, in addition to those of the parents, to determine a universal set. In the MLP example, the universal set of the dropout ρ reduces to $\overline{P} = [0, 0.5]$, *i.e.*, it is obtained when $\sum_{i=1}^{l} u_i = \tau_{\max}$. Recall that, from (1), the bounds of the dropout ρ depends on its parents the number of hidden layers l and the number of units u_i. However, to determine its universal set $\overline{P}$, the constant $\tau_{\max} = \max\left\{\sum_{i=1}^{l} u_i \ : \ o \in \{\texttt{ASGD}, \texttt{ADAM}\},\ l \in L_o,\ u_i \in U_o \text{ for } 1 \leq i \leq l\right\}$ must be determined by considering the optimizer o (grandparent). Again, τ is a constant, hence the dropout ρ has no decree dependency with the optimizer o.

The next step is to develop the restricted set of a variable. The restricted set of the variable $\overline{x}_i^r$ is the subset of the universal set $\overline{\mathcal{X}}_i^r$ such that $\overline{x}_i^r$ respects the decree dependencies for the given values of its parents $\overline{\text{par}}_i^r$, which are identified by arcs of the role graph G. The formal definition of the restricted set is presented below.

Definition 10 (Restricted set)**.** *For $r \in R$ and $i \in I^r$, the restricted set of the variable $\overline{x}_i^r$ is the universal set $\overline{\mathcal{X}}_i^r$ conditioned by the values of its parents $\overline{\text{par}}_i^r$, expressed as*

$$\overline{\mathcal{X}}_i^r/_{\overline{\text{par}}_i^r} := \left\{\overline{x}_i^r \in \overline{\mathcal{X}}_i^r \ : \ \forall\ (\overline{v}, \overline{x}_i^r) \in A,\ \overline{x}_i^r \textit{ respects the decree dependencies for the values of } \overline{v}\right\}.$$

Meta and neutral variables are always included and have no parent, hence their restricted set is simply their universal set: $\overline{\mathcal{X}}_i^{\text{m}}/_{\overline{\text{par}}_i^{\text{m}}} = \overline{\mathcal{X}}_i^{\text{m}}\ \forall i \in I^{\text{m}}$ and $\overline{\mathcal{X}}_j^{\text{neu}}/_{\overline{\text{par}}_j^{\text{neu}}} = \overline{\mathcal{X}}_j^{\text{neu}}\ \forall j \in I^{\text{neu}}$. The restricted set of a meta-decreed or decreed variable requires the values of its parents for determining both its inclusion and its admissible values when it is included. If the decree dependencies, given the values of parent variables, dictate that a child is excluded, then its restricted set is $\{\texttt{EXC}\}$. In the MLP example, the restrictive set of $\overline{\alpha}_1$ is $\{\texttt{EXC}\}$ when the optimizer $\overline{o} = \texttt{ADAM}$. The notation for restricted sets uses a diagonal bar, instead of a vertical bar commonly used in probability and statistics, to avoid any confusion with the logical OR operator or abbreviation of "*such that*". Furthermore, the parents are placed as subscripts to outline the dependency.

3.5 Extended domain and transfer mapping

Now that the restricted sets have been detailed, the extended domain $\overline{\mathcal{X}}$ is formally introduced.

Definition 11 (Extended domain)**.** *The extended domain $\overline{\mathcal{X}}$ is a graph-structured domain constructed from a domain $\mathcal{X}$, its corresponding role graph $G = (V, A)$ and from the restricted sets of all included and excluded variables. The extended domain $\overline{\mathcal{X}}$ is expressed as*

$$\overline{\mathcal{X}} := \left\{ \; \overline{x} \;\; : \;\; \overline{\mathcal{X}}^r_i /_{\overline{\text{par}}^r_i}, \;\; \forall r \in R \; \forall i \in I^r \; \right\}.$$

Definition 11 expresses that an extended point $\overline{x} \in \overline{\mathcal{X}}$ must respect all the inclusion-exclusion or admissible values dependencies, *i.e.*, decree dependencies, between its variables. Recall that the main objective of this work is to equip the domain $\mathcal{X}$ with a distance $\text{dist}_p : \mathcal{X} \times \mathcal{X} \to \overline{\mathbb{R}}^+$ to facilitate optimization or machine learning tasks on domains that involve heterogeneous dataset. This will be done by introducing the graph-structured distance $\overline{\text{dist}}_p : \overline{\mathcal{X}} \times \overline{\mathcal{X}} \to \overline{\mathbb{R}}^+$ on the extended domain $\overline{\mathcal{X}}$, and then by inducing a distance on the domain via the bijective mapping defined in the following theorem.

Theorem 2 (One-to-one correspondence)**.** *The transfer mapping $T_G : \mathcal{X} \to \overline{\mathcal{X}}$, which assigns the extended point $T_G(x) \in \overline{\mathcal{X}}$ to any point $x \in \mathcal{X}$ by adding its excluded variables determined by the role graph G, is a bijection.*

Proof. Injectivity. Let $x, y \in \mathcal{X}$ be such that $x \neq y$. By definition of T_G, there is a least one variable whose value differ between the two extended points $\overline{x} = T_G(x)$ and $\overline{y} = T_G(y)$, since 1) there is a least one included variable between x and y that does not share the same value; or 2) there is a least one variable that is strictly excluded for one point between x and y. Therefore, there is a least one variable that also does not share the same value in the extended points $\overline{x}$ and $\overline{y}$. This show that T_G is injective since $x \neq y \Rightarrow T_G(x) \neq T_G(y)$.

Surjectivity. Let $\overline{x} \in \overline{\mathcal{X}}$ be an extended point. Then, let $G' = (V', E')$ be the subgraph of the role graph G obtained by removing the nodes and arcs corresponding to the excluded variables that take the value `EXC` in $\overline{x}$. The set V' is nonempty since either $\overline{x}$ has no meta variables and thus $V' = V$, or $\overline{x}$ has at least one meta variable $\overline{x}^{\text{m}}_i = x^{\text{m}}_i \in V'$. Thus, V' is the set of variables that are included in $\overline{x}$, and E' is the set of decree dependencies between the included variables of $\overline{x}$. By construction, each variable in V' respects the decree dependencies between the other variables in the set V'. Let x be the point that contains only the included variables of the extended point $\overline{x}$, *i.e.*, the variables in the set V'. The point x necessarily belongs to domain $\mathcal{X}$, since it only contains included variables that respect the decree dependencies between each others. Indeed, otherwise, if the decree dependencies would not be respected, then x would 1) contain a variable that should not be included; or 2) not contain a variable that should be included; or 3) contain an included variable that would take a value that is not allowed by the decree dependencies. Then, by definition of the mapping T_G, $T_G(x) = \overline{x}$, since $x \in \mathcal{X}$ contains all the included variables of the extended point $\overline{x} \in \overline{\mathcal{X}}$. This shows that T_G is thus surjective, since $\forall \overline{x} \in \overline{\mathcal{X}}$, $\exists x \in \mathcal{X}$, such that $T_G(x) = \overline{x}$.

The transfer mapping T_G is both injective and surjective, thus bijective. □

A consequence of Theorem 2 is that if a distance $\overline{\text{dist}}_p : \overline{\mathcal{X}} \times \overline{\mathcal{X}} \to \overline{\mathbb{R}}^+$ is well-defined on the extended domain $\overline{\mathcal{X}}$, then a distance $\text{dist}_p : \mathcal{X} \times \mathcal{X} \to \overline{\mathbb{R}}^+$ can be induced on the domain $\mathcal{X}$ with the bijective transfer mapping $T_G : \mathcal{X} \to \overline{\mathcal{X}}$. This is done in Section 4.

3.6 Illustration on the MLP example: graph-structured domain

Section 3 introduced many definitions and mathematical expressions. To facilitate the comprehension, the MLP example is modeled in this section. Before developing the role graph, the preliminary step is to model each variable and its set, which takes into account its decree dependencies, for a point $x \in \mathcal{X}$. Variables are modeled by following the decree dependencies, starting with the meta and neutral variables, then modeling variables with decree dependencies whose determining variables have been previously modeled.

The first variables to be modeled are the optimizer $o \in \{\texttt{ASGD}, \texttt{ADAM}\}$, the learning rate $r \in]0,1[$, and the activation function $a \in \{\texttt{ReLU}, \texttt{Sig}, \texttt{Tanh}\}$. Next, meta-decreed and decreed variables whose decree dependencies involve only meta variables are modeled. The variables α_1, α_2 and α_3 are included if and only if $o = \texttt{ASGD}$, and the variables β_1, β_2 and β_3 are included if and only if $o = \texttt{ADAM}$. The number l of hidden layers is constrained to a set that depends on the optimizer o:

$$l \in L_o = \begin{cases} L_{\texttt{ASGD}} = \{0,1,2,3\} & \text{if } o = \texttt{ASGD}, \\ L_{\texttt{ADAM}} = \{0,1,2,3,4\} & \text{if } o = \texttt{ADAM}. \end{cases}$$

The number of units u_i is included if $1 \leq i \leq l \leq l_{\max} = 4$ and its bounds depends on the optimizer o:

$$u_i \in U_o = \begin{cases} U_{\texttt{ASGD}} & \text{if } o = \texttt{ASGD}, \\ U_{\texttt{ADAM}} & \text{if } o = \texttt{ADAM}. \end{cases}$$

There are at most four unit variables in the architecture. Finally, the dropout ρ is always included, but its bounds depends on the number of the hidden layers l and the number of units $u_1, u_2, \ldots, u_l$ as expressed in (1).

The next modeling step is to build the role graph $G = (V, A)$. The nodes of the graph correspond to all included and excluded variables $V = \{\overline{o}, \overline{\alpha}_1, \overline{\alpha}_2, \overline{\alpha}_3, \overline{\beta}_1, \overline{\beta}_2, \overline{\beta}_3, \overline{l}, \overline{u}_1, \overline{u}_2, \overline{u}_3, \overline{u}_4, \overline{\rho}, \overline{r}, \overline{a}\}$. The arcs A of the graph represent pairs of parent-child variables, in which the child has a decree dependency with its parent. For example, the number of hidden layers l and the number of units u_i are all parents of the dropout ρ, since they determine its bounds. The role graph $G = (V, A)$ of the MLP example is illustrated in Figure 3. The role graph G schematically models a great deal of information, including 1) all included and excluded variables, 2) the role of each variable via their node position, 3) the dependencies of the restricted sets of meta-decreed and decreed variables with their parents.

The third modeling step is to determine the universal sets of meta-decreed and decreed variables. The universal sets of meta and neutral variables are directly given by the problem statement. The universal set of the number of hidden layers $\overline{l}$, the dropout ρ, and the number of units $\overline{u}_i$ are respectively $\overline{L} = L_{\texttt{ASGD}} \cup L_{\texttt{ADAM}}$, $\overline{P} = [0, 0.5]$, and $\overline{U}_i = U_{\texttt{ASGD}} \cup U_{\texttt{ADAM}} \cup \{\texttt{EXC}\}$ for $1 \leq 1 \leq l_{\max}$.

Afterwards, the fourth modeling step is to express the restricted sets $\overline{\mathcal{X}}_i^r/_{\overline{\mathrm{par}}_i^r}$. The restricted sets of meta and neutral variables are identical to their universal set $\overline{\mathcal{X}}_i^r$, which can be expressed as

$$\overline{x}_1^{\mathrm{m}} = \overline{o} \in \{\texttt{ASGD}, \texttt{ADAM}\}, \qquad \text{and} \qquad (\overline{x}_1^{\mathrm{neu}}, \overline{x}_2^{\mathrm{neu}}) = (\overline{r}, \overline{a}) \in]0,1[\times \{\texttt{ReLU}, \texttt{Sig}, \texttt{Tanh}\}.$$

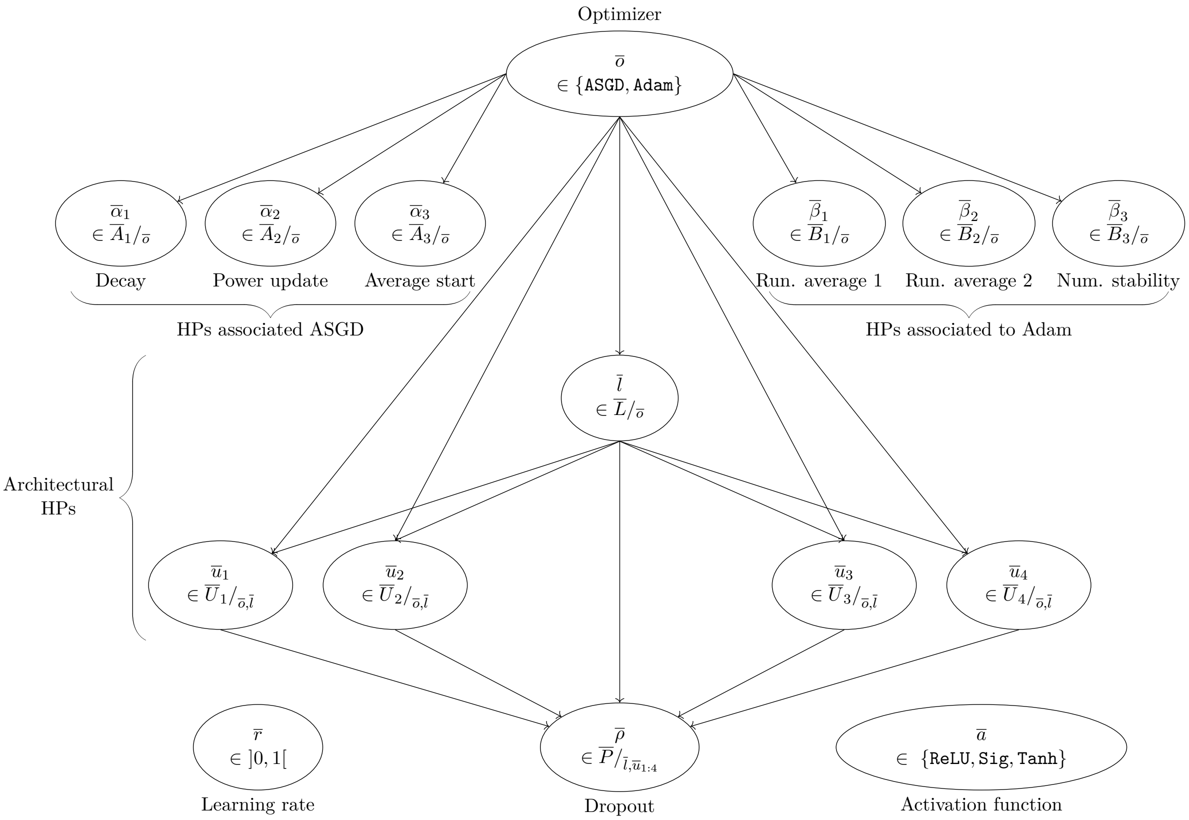
Optimizer
$\overline{o}$
$\in \{\texttt{ASGD}, \texttt{Adam}\}$
$\overline{\alpha}_1$
$\in \overline{A}_1/_{\overline{o}}$
$\overline{\alpha}_2$
$\in \overline{A}_2/_{\overline{o}}$
$\overline{\alpha}_3$
$\in \overline{A}_3/_{\overline{o}}$
$\overline{\beta}_1$
$\in \overline{B}_1/_{\overline{o}}$
$\overline{\beta}_2$
$\in \overline{B}_2/_{\overline{o}}$
$\overline{\beta}_3$
$\in \overline{B}_3/_{\overline{o}}$
Decay
Power update
Average start
Run. average 1
Run. average 2
Num. stability
HPs associated ASGD
HPs associated to Adam
$\overline{l}$
$\in \overline{L}/_{\overline{o}}$
Architectural HPs
$\overline{u}_1$
$\in \overline{U}_1/_{\overline{o},\overline{l}}$
$\overline{u}_2$
$\in \overline{U}_2/_{\overline{o},\overline{l}}$
$\overline{u}_3$
$\in \overline{U}_3/_{\overline{o},\overline{l}}$
$\overline{u}_4$
$\in \overline{U}_4/_{\overline{o},\overline{l}}$
$\overline{r}$
$\in]0,1[$
Learning rate
$\overline{\rho}$
$\in \overline{P}/_{\overline{l},\overline{u}_{1:4}}$
Dropout
$\overline{a}$
$\in \{\texttt{ReLU}, \texttt{Sig}, \texttt{Tanh}\}$
Activation function

Figure 3: Role graph G for the MLP example.

Meta-decreed variables and their restricted sets are expressed as

$$\left(\overline{x}_1^{\text{md}}, \overline{x}_2^{\text{md}}, \overline{x}_3^{\text{md}}, \overline{x}_4^{\text{md}}, \overline{x}_5^{\text{md}}\right) = \left(\overline{l}, \overline{u}_{1:4}\right) \in \overline{L}/_{\overline{o}} \times \prod\nolimits_{i=1}^{4} \overline{U}_i/_{\overline{o},\overline{l}}$$

where $\overline{u}_{1:4} := (\overline{u}_1, \overline{u}_2, \overline{u}_3, \overline{u}_4)$, and

$$\overline{L}/_{\overline{o}} = \begin{cases} L_{\texttt{ASGD}} & \text{if } \overline{o} = \ \texttt{ASGD}, \\ L_{\texttt{ADAM}} & \text{if } \overline{o} = \ \texttt{ADAM}, \end{cases} \qquad \text{and} \qquad \overline{U}_i/_{\overline{o},\overline{l}} = \begin{cases} U_{\texttt{ASGD}} & \text{if } \overline{o} = \texttt{ASGD} \text{ and } 1 \leq i \leq \overline{l}, \\ U_{\texttt{ADAM}} & \text{if } \overline{o} = \texttt{ADAM} \text{ and } 1 \leq i \leq \overline{l}, \\ \{\texttt{EXC}\} & \text{otherwise}, \end{cases}$$

with $1 \leq i \leq \overline{l}$. Decreed variables are expressed as

$$\left(\overline{x}_1^{\text{dec}}, \overline{x}_2^{\text{dec}}, \overline{x}_3^{\text{dec}}, \overline{x}_4^{\text{dec}}, \overline{x}_5^{\text{dec}}, \overline{x}_6^{\text{dec}}, \overline{x}_7^{\text{dec}}\right) = \left(\overline{\alpha}_{1:3}, \overline{\beta}_{1:3}, \overline{\rho}\right) \in \prod\nolimits_{i=1}^{3} \overline{A}_i/_{\overline{o}} \times \prod\nolimits_{j=1}^{3} \overline{B}_i/_{\overline{o}} \times \overline{P}/_{\overline{l},\overline{u}_{1:4}},$$

where $\overline{\alpha}_{1:3} := (\overline{\alpha}_1, \overline{\alpha}_2, \overline{\alpha}_3)$, $\overline{\beta}_{1:3} := \left(\overline{\beta}_1, \overline{\beta}_2, \overline{\beta}_3\right)$, and

$$\overline{A}_i/_{\overline{o}} = \begin{cases}]0,1[& \text{if } \overline{o} = \texttt{ASGD}, \\ \{\texttt{EXC}\} & \text{if } \overline{o} = \texttt{ADAM}, \end{cases} \qquad \text{and} \qquad \overline{B}_j/_{\overline{o}} = \begin{cases} \{\texttt{EXC}\} & \text{if } \overline{o} = \texttt{ASGD}, \\ ]0,1[& \text{if } \overline{o} = \texttt{ADAM}, \end{cases}$$

with $i, j \in \{1, 2, 3\}$, and

$$\overline{P}/_{\overline{l},\overline{u}_{1:4}} = \left[0, \ \frac{\sum_{i=1}^{\overline{l}} \overline{u}_i}{2\tau_{\max}}\right] \subseteq [0, 0.5]. \tag{5}$$

Finally, the extended domain $\overline{\mathcal{X}}$ is constructed with the restricted sets as follows

$$\overline{\mathcal{X}} = \left\{ \begin{array}{lllll} \overline{x} & : & \overline{o} & \in & \{\texttt{ASGD}, \texttt{ADAM}\}, \\ & & (\bar{l}, \overline{u}_{1:4}) & \in & \overline{L}/_{\overline{o}} \times \prod_{i=1}^{4} \overline{U}_i/_{\overline{o},\bar{l}} \\ & & (\overline{\alpha}_{1:3}, \overline{\beta}_{1:3}, \overline{\rho}) & \in & \prod_{i=1}^{3} \overline{A}_i/_{\overline{o}} \times \prod_{j=1}^{3} \overline{B}_i/_{\overline{o}} \times \overline{P}/_{\bar{l},\overline{u}_{1:4}}, \\ & & (\overline{r}, \overline{a}) & \in &]0,1[\ \times \{\texttt{ReLU}, \texttt{Sig}, \texttt{Tanh}\} \end{array} \right\}.$$

4 Distance for graph-structured domains

In this section, the graph-structured distance $\overline{\text{dist}}_p : \overline{\mathcal{X}} \times \overline{\mathcal{X}} \to \overline{\mathbb{R}}^+$ is defined. Section 4.1 presents the included-excluded distance function that can compute distances for variables that can be included or excluded. The included-excluded distances of the MLP example are modeled in Section 4.2. Finally, Section 4.3 presents the graph-structured distance that is constructed with included-excluded distances, one per variable.

4.1 Included-excluded distances

For two extended points $\overline{x}, \overline{y} \in \overline{\mathcal{X}}$, the distance regarding the i-th variable assigned to role r, respectively $\overline{x}_i^r$ and $\overline{y}_i^r$, is computed through three cases:

1. both $\overline{x}_i^r$ and $\overline{y}_i^r$ are excluded, hence the distance is set to zero;
2. exactly one variable $\overline{x}_i^r$ or $\overline{y}_i^r$ is excluded, hence the distance is set to a parameter that models a distance between a variable that is included for one extended point, and excluded for the other extended point;
3. both $\overline{x}_i^r$ and $\overline{y}_i^r$ are included, hence an one-dimensional distance function d is used, *e.g.*, the Euclidean distance.

Recall that a meta or neutral variable $\overline{x}_i^r \in \overline{\mathcal{X}}_i^r$ is always included, hence for $r \in \{\text{m}, \text{neu}\}$, the distance between $\overline{x}_i^r$ and $\overline{y}_i^r$ is always computed in the third case. For a meta-decreed or decreed variable, the restricted sets of $\overline{x}_i^r$ and $\overline{y}_i^r$ may differ, hence its corresponding included-excluded distance must be defined on its universal set $\overline{\mathcal{X}}_i^r$ in order to allow comparisons of any pairs $\overline{x}_i^r, \overline{y}_i^r$ with different restricted sets. In the MLP example, the universal set of the number of units $\overline{u}_i$ is $\overline{U} = U_{\texttt{ASGD}} \cup U_{\texttt{ADAM}} \cup \{\texttt{EXC}\}$. Hence, to compare the number of units $\overline{u}_i$ from any two pairs of extended points, the corresponding included-excluded distance of $\overline{u}_i$ must be defined on its universal set $\overline{U}$. The following theorem formalizes the discussion above on the three cases and the universal set by introducing a novel distance based on distances proposed in [34, 32].

Theorem 3 (Included-excluded distance)**.** *Let $\overline{\mathcal{X}}_i^r$ be the universal set of the i-th variable assigned to the role $r \in R$, noted $\overline{x}_i^r$, and define $\overline{\mathcal{Y}}_i^r = \overline{\mathcal{X}}_i^r \setminus \{\texttt{EXC}\}$. Consider $d : \overline{\mathcal{Y}}_i^r \times \overline{\mathcal{Y}}_i^r \to \overline{\mathbb{R}}^+$, a one-dimensional extended real-valued distance for the variable $\overline{x}_i^r$ when it is included,* *and* $\theta_i^r \in \overline{\mathbb{R}}^+$ *a parameter greater than or equal to* $\sup\{d\,(\mu,\nu) : \mu,\nu \in \overline{\mathcal{Y}}_i^r\}/2$*. Then, for* $u,v \in \overline{\mathcal{X}}_i^r$*, the function* $d_i^r : \overline{\mathcal{X}}_i^r \times \overline{\mathcal{X}}_i^r \to \overline{\mathbb{R}}^+$ *defined by*

$$d_i^r\,(u,v) := \begin{cases} d\,(u,v) & \textit{if} \;\; u \neq \texttt{EXC} \neq v \;\; \textit{(both included)}, \\ 0 & \textit{if} \;\; u = \texttt{EXC} = v \;\; \textit{(both excluded)}, \\ \theta_i^r & \textit{otherwise} \qquad\quad \textit{(one excluded)}, \end{cases} \tag{6}$$

is a one-dimensional extended real-valued distance function.

Proof. Let $r \in R$ and $i \in I^r$. The identity of indiscernibles, nonnegativity and symmetry of d_i^r are trivially proven since θ_i^r is strictly positive and since d is a distance function. The rest of the proof consists of proving that d_i^r satisfies the triangle inequality. Let $u,v,z \in \overline{\mathcal{X}}_i^r$:

Case 1 (both variables are excluded): if $u = \texttt{EXC} = v$, then

$$d_i^r(u,v) = 0 \;\leq\; d_i^r(u,z) \;+\; d_i^r(z,v), \text{ by nonnegativity of } d_i^r.$$

Case 2 (only one variable is excluded, WLOG v): if $u \neq \texttt{EXC} = v$ and

- if $z \neq \texttt{EXC}$, then

$$d_i^r(u,v) = \theta_i^r \;\leq\; d(u,z) \;+\; \theta_i^r \;=\; d_i^r(u,z) \;+\; d_i^r(z,v), \text{ by nonnegativity of } d.$$

- if $z = \texttt{EXC}$, then

$$d_i^r(u,v) = \theta_i^r \;\leq\; \theta_i^r \;+\; 0 \;=\; d_i^r(u,z) \;+\; d_i^r(z,v).$$

Case 3 (both variables are included): if $u \neq \texttt{EXC}$ and $v \neq \texttt{EXC}$ and

- if $z \neq \texttt{EXC}$, then

$$d_i^r(u,v) \;=\; d(u,v) \;\leq\; d(u,z) \;+\; d(z,v) \;=\; d_i^r(u,z) \;+\; d_i^r(z,v).$$

- if $z = \texttt{EXC}$, then

$$d_i^r(u,v) \;=\; d(u,v) \;\leq\; \sup\{d\,(\mu,\nu) : \mu,\nu \in \overline{\mathcal{Y}}_i^r\} \;\leq\; 2\theta_i^r = d_i^r(u,z) \;+\; d_i^r(z,v).$$

□

Multiple comments on Theorem 3 that introduces the included-excluded distance are provided. First, the included-excluded distance is precisely useful for meta-decreed and decreed variables that can have different restricted sets, *i.e.* that can be included or excluded, and/or have different admissible values. As mentioned previously, for variables that are always included, such as meta and neutral variables, included-excluded distances are always computed with the both included case. The included-excluded distance is nevertheless defined for these variables in order to obtain a concise formulation of the graph-structured distance in Section 4.3.

Second, the included-excluded distance is compatible with any variable type. Indeed, in (6), the two cases both excluded and one excluded does not regard the variable type, and the case both included allows to utilize any distance function d.

Third, the universal set $\overline{\mathcal{X}}_i^r$ allows to compare any pair of variables $\overline{x}_i^r, \overline{y}_i^r$. Moreover, it is also necessary to establish a lower bound on the parameter θ_i^r, which ensures the triangular inequality in the last case of the proof. The inequality $\theta_i^r \geq \sup\{d\left(\mu, \nu\right) : \mu, \nu \in \overline{\mathcal{X}}_i^r \setminus \{\texttt{EXC}\}\}/2$ implies that the distance $d_i^r(\overline{x}_i^r, \overline{y}_i^r) = \theta_i^r$ (one excluded case) must be at least half the largest distance between any pairs of included variables $\overline{x}_i^r, \overline{y}_i^r$, with possibly different restricted sets. In the MLP example, recall that the universal set of the number of units $\overline{u}_i$ is $\overline{U} = U_{\texttt{ASGD}} \cup U_{\texttt{ADAM}} \cup \{\texttt{EXC}\}$. Therefore, the parameter θ_i^r for $\overline{u}_i$ must be greater than $(\max(\overline{U}') - \min(\overline{U}'))/2$, where $\overline{U}' = U_{\texttt{ASGD}} \cup U_{\texttt{ADAM}}$. Apart from its lower bound, the parameter θ_i^r is flexible.

Fourth, the included-excluded distance is more formally an extended real-valued one [10], since it is allowed to take the infinite value. The infinity value allows to consider meta-decreed or decreed variables with unbounded restricted sets. For example, let $r \in \{\mathrm{md}, \mathrm{dec}\}$ and $i \in I^r$, such that $\overline{\mathcal{X}}_i^r/_{\overline{\mathrm{par}}_i^r} = [0, \infty[$ when it is included, and $\overline{\mathcal{X}}_i^r/_{\overline{\mathrm{par}}_i^r} = \{\texttt{EXC}\}$ when it is excluded. In this example, $\overline{\mathcal{X}}_i^r = [0, \infty[\, \cup \, \{\texttt{EXC}\}$, therefore $\sup\{d\left(\mu, \nu\right) : \mu, \nu \in \overline{\mathcal{X}}_i^r \setminus \{\texttt{EXC}\}\} = \infty$, hence the parameter θ_i^r must be set to infinity to guaranty the triangular inequality. If the restricted sets are always bounded, then included-excluded distance becomes a standard distance that maps into $\mathbb{R}$, instead of $\overline{\mathbb{R}}$.

4.2 Illustration on the MLP example: distances

In this section, the included-excluded distances of each variable in the MLP example are modeled. For a meta-decreed or decreed variable that can be excluded, its parameter is arbitrarily set to $\theta_i^r = \frac{3}{2} \sup\{d\left(\mu, \nu\right) \ : \ \mu, \nu \in \overline{\mathcal{X}}_i^r \setminus \{\texttt{EXC}\}\}$.

The distance for the optimizer, which is the only meta variable, is

$$d_1^m\left(\overline{o}, \overline{o}'\right) = \begin{cases} \sigma & \text{if } \overline{o} \neq \overline{o}', \\ 0 & \text{otherwise}, \end{cases} \tag{7}$$

where $\sigma > 0$ is a categorical parameter. Similarly, the distances for the neutral variables are

$$d_1^{\mathrm{neu}}\left(\overline{r}, \overline{r}'\right) = \left|\overline{r} - \overline{r}'\right|, \text{ and } d_2^{\mathrm{neu}}\left(\overline{a}, \overline{a}'\right) = \begin{cases} \gamma_1 \text{ if } (\overline{a} = \texttt{ReLU}, \, \overline{a}' = \texttt{Sig}) \text{ or } (\overline{a} = \texttt{Sig}, \, \overline{a}' = \texttt{ReLU}), \\ \gamma_2 \text{ if } (\overline{a} = \texttt{ReLU}, \, \overline{a}' = \texttt{Tanh}) \text{ or } (\overline{a} = \texttt{Tanh}, \, \overline{a}' = \texttt{ReLU}), \\ \gamma_3 \text{ if } (\overline{a} = \texttt{Sig}, \, \overline{a}' = \texttt{Tanh}) \text{ or } (\overline{a} = \texttt{Tanh}, \, \overline{a}' = \texttt{Sig}), \\ \ 0 \text{ if } \overline{a} = \overline{a}', \end{cases}$$

where $\gamma_1, \gamma_2, \gamma_3 > 0$ are categorical parameters. For meta-decreed variables, the included-excluded distances are

$$d_1^{\mathrm{md}}\left(\overline{l}, \overline{l}'\right) = |l - l'|$$

and

$$d_{i+1}^{\mathrm{md}}\left(\overline{u}_i, \overline{u}_i'\right) = \begin{cases} 0 & \text{if } \overline{l} < i \text{ and } \overline{l}' < i, \\ \dfrac{3}{2}\left(\max\left(\overline{U}'\right) - \min\left(\overline{U}'\right)\right) & \text{if } \left(\overline{l} \geq i \text{ and } \ \overline{l}' < i\right) \text{ or } \left(\overline{l} < i \text{ and } \overline{l}' \geq i\right), \\ \left|\overline{u}_i - \overline{u}_i'\right| & \text{if } \overline{l} \geq i \text{ and } \overline{l}' \geq i, \end{cases}$$

where $i \in \{1,2,\dots,l_{\max}\}$, $\overline{U}' = U_{\texttt{ASGD}} \cup U_{\texttt{ADAM}}$, and $\overline{l} < i$ signifies that the i-th unit is not included, whereas $\overline{l} \geq i$ signifies that it is included. Finally, the included-excluded distances for the decreed variables are given by the expressions

$$d_i^{\text{dec}}\left(\overline{\alpha}_i, \overline{\alpha}_i'\right) = \begin{cases} 0 & \text{if } \overline{o} = \texttt{ADAM} = \overline{o}', \\ \dfrac{3}{2} & \text{if } \overline{o} \neq \overline{o}', \\ \left|\overline{\alpha}_i - \overline{\alpha}_i'\right| & \text{if } \overline{o} = \texttt{ASGD} = \overline{o}', \end{cases} \quad \text{and} \quad d_{j+3}^{\text{dec}}\left(\overline{\beta}_j, \overline{\beta}_j'\right) = \begin{cases} 0 & \text{if } \overline{o} = \texttt{ASGD} = \overline{o}', \\ \dfrac{3}{2} & \text{if } \overline{o} \neq \overline{o}', \\ \left|\overline{\beta}_j - \overline{\beta}_j'\right| & \text{if } \overline{o} = \texttt{ADAM} = \overline{o}', \end{cases}$$

where $i, j \in \{1,2,3\}$, and

$$d_7^{\text{dec}}\left(\overline{\rho}, \overline{\rho}'\right) = \left|\overline{\rho} - \overline{\rho}'\right|.$$

4.3 Graph-structured distance and induced distance

Now that the included-excluded distance has been detailed, the following theorem formally introduces the graph-structured distance.

Theorem 4 (Graph-structured distance)**.** *For any $p \geq 1$, the graph-structured function $\overline{\text{dist}}_p : \overline{\mathcal{X}} \times \overline{\mathcal{X}} \to \overline{\mathbb{R}}^+$ defined by*

$$\overline{\text{dist}}_p(\overline{x}, \overline{y}) := \left(\sum_{r \in R} \sum_{i \in I^r} d_i^r \left(\overline{x}_i^r, \overline{y}_i^r\right)^p \right)^{\frac{1}{p}}, \tag{8}$$

is an extended real-valued distance function, where $R = \{\text{m}, \text{md}, \text{dec}, \text{neu}\}$.

Proof. The identity of indiscernibles, nonnegativity and symmetry of $\overline{\text{dist}}_p$ are trivially proven since the operations of summation and exponentiation with $p \geq 1$ on the included-excluded distances in (8) conserve these properties. The rest of the proof consists of showing the triangular inequality is respected by demonstrating that $\overline{\text{dist}}_p$ is equivalent to a p-norm, that respects the triangular inequality. Let $K = \{1, 2, \dots, n^{\text{m}} + n^{\text{md}} + n^{\text{dec}} + n^{\text{neu}}\}$ be a set of indices that reorders the indices $r \in \{\text{m}, \text{md}, \text{dec}, \text{neu}\}$ and $i \in \{1, 2, \dots, n^r\}$:

- $a_k := d_i^{\text{m}}\left(\overline{x}_i^{\text{m}}, \overline{y}_i^{\text{m}}\right)$, for $k = i$ with $i \in \{1, 2, \dots, n^{\text{m}}\}$,
- $a_k := d_j^{\text{md}}\left(\overline{x}_j^{\text{md}}, \overline{y}_j^{\text{md}}\right)$, for $k = n^{\text{m}} + j$ with $j \in \{1, 2, \dots, n^{\text{md}}\}$,
- $a_k := d_l^{\text{dec}}\left(\overline{x}_l^{\text{dec}}, \overline{y}_l^{\text{dec}}\right)$, for $k = n^{\text{m}} + n^{\text{md}} + l$ with $l \in \{1, 2, \dots, n^{\text{dec}}\}$,
- $a_k := d_v^{\text{neu}}\left(\overline{x}_v^{\text{neu}}, \overline{y}_v^{\text{neu}}\right)$, for $k = n^{\text{m}} + n^{\text{md}} + n^{\text{dec}} + v$ with $v \in \{1, 2, \dots, n^{\text{neu}}\}$.

Finally, let $a = \left(a_1, a_2, \dots, a_{|K|}\right)$, then

$$\|a\|_p = \left(\sum_{k=1}^{|K|} |a_k|^p \right)^{\frac{1}{p}} = \left(\sum_{r \in R} \sum_{i \in I^r} d_i^r \left(\overline{x}_i^r, \overline{y}_i^r\right)^p \right)^{\frac{1}{p}} = \overline{\text{dist}}_p\left(\overline{x}, \overline{y}\right)$$

□

For $p \to \infty$, the graph-structured distance $\overline{\mathrm{dist}}_\infty : \overline{\mathcal{X}} \times \overline{\mathcal{X}} \to \overline{\mathbb{R}}^+$ is defined as a maximum, *i.e.*,

$$\overline{\mathrm{dist}}_\infty \left(\overline{x}, \overline{y}\right) := \max \left\{ \, d_i^r \left(\overline{x}_i^r, \overline{y}_i^r\right) \; : \; r \in R, \; i \in I^r \right\}, \tag{9}$$

which is trivially a distance function by virtue of the max function.

Note that, in practice, variables often require scaling to improve the conditioning and eliminate biases related to variable scales. In the context of the work, scaling categorical variables and excluded variables is ambiguous. Fortunately, in our proposed distance, an included-excluded distance d_i^r is defined with a flexible distance d for its both included case. Therefore, the distance d can be defined using a scaling parameter, such that $d(a, b) = \omega_i^r d'(a, b)$, where $\omega_i^r > 0$ is a weight parameter related to the variable $\overline{x}_i^r$ and d' is a one-dimensional distance of appropriate variable type. The weight parameter ω_i^r can be used to automatically scale the lower bounds of the parameter θ_i^r, since the lower bound of θ_i^r is defined with the one-dimensional distance d. In Section 5, scaling parameters will be used to better adjust our proposed distance to the datasets; this helps to remove biases that are related to variable scales.

Theorem 2, which establishes the bijection between the domain $\mathcal{X}$ and the extended domain $\overline{\mathcal{X}}$, implies that a distance $\mathrm{dist}_p : \mathcal{X} \times \mathcal{X} \to \overline{\mathbb{R}}^+$ can be induced from the graph-structured distance $\overline{\mathrm{dist}}_p : \overline{\mathcal{X}} \times \overline{\mathcal{X}} \to \overline{\mathbb{R}}^+$. The following corollary is a direct consequence of Theorems 2 and 4.

Corollary 1 (Induced distance)**.** *For $p \geq 1$, the induced function $\mathrm{dist}_p : \mathcal{X} \times \mathcal{X} \to \overline{\mathbb{R}}^+$ defined by*

$$\mathrm{dist}_p(x, y) := \overline{\mathrm{dist}}_p \left(T_G(x), T_G(y)\right) = \overline{\mathrm{dist}}_p \left(\overline{x}, \overline{y}\right), \tag{10}$$

is an extended real-valued distance, where $\overline{\mathrm{dist}}_p : \overline{\mathcal{X}} \times \overline{\mathcal{X}} \to \overline{\mathbb{R}}^+$ is a graph-structured distance and $T_G : \mathcal{X} \to \overline{\mathcal{X}}$ is the bijective transfer mapping.

Corollary 1 unpacks most of the contributions. To arrive at Corollary 1, it was necessary to: 1) define an extended point $\overline{x}$, restricted sets and the extended domain $\overline{\mathcal{X}}$ using notions from graph theory; 2) define the transfer mapping $T_G : \mathcal{X} \to \overline{\mathcal{X}}$, and prove that it is bijective, 3) define the included-excluded distances on the universal set for tackling variables that can either be included or excluded, or with different admissible values, and 4) define the graph-structured distance $\overline{\mathrm{dist}}_p : \overline{\mathcal{X}} \times \overline{\mathcal{X}} \to \overline{\mathbb{R}}^+$ based on the contributions mentioned above.

5 Computational experiments involving heterogeneous datasets

This section compares two approaches on regression problems with simple distance-based models for instances of the MLP example. The first approach, called Sub divides a regression problem into subproblems, each assigned to a portion of the domain in which the included variables are fixed. The second approach, called Graph, uses the induced distance in (10) with $p = 2$ to tackle a regression problem directly and by aggregating the data across the subproblems. See this git-link for full details on the computational experiments. Table 1 details the instances for the two approaches.

	Sub					Graph			
Instance	o	l	Variables	# of var.	Size	Variables	# of var.	# of θ_i^r	Size
#1	ASGD	1	$r,\quad u_1$	2	40	$l, r, u_{1:3}$	5	2	180
		2	$r,\quad u_{1:2}$	3	60				
		3	$r,\quad u_{1:3}$	4	80				
#2	ASGD	1	$r,\quad u_1,\quad \alpha_{1:3}$	5	100	$l, r, u_{1:3}, \alpha_{1:3}$	8	2	360
		2	$r,\quad u_{1:2},\quad \alpha_{1:3}$	6	120				
		3	$r,\quad u_{1:3},\quad \alpha_{1:3}$	7	140				
#3	ASGD	1	$r,\quad u_1,\quad \alpha_{1:3}$	5	100	$o, l, r, u_{1:2}, \alpha_{1:3}, \beta_{1:3}$	11	7	440
		2	$r,\quad u_{1:2},\quad \alpha_{1:3}$	6	120				
	ADAM	1	$r,\quad u_1,\quad \beta_{1:3}$	5	100				
		2	$r,\quad u_{1:2},\quad \beta_{1:3}$	6	120				
#4	ASGD	1	$r,\quad u_1,\quad \alpha_{1:3}$	5	100	$o, l, r, u_{1:3}, \alpha_{1:3}, \beta_{1:3}$	12	8	580
		2	$r,\quad u_{1:2},\quad \alpha_{1:3}$	6	120				
	ADAM	1	$r,\quad u_1,\quad \beta_{1:3}$	5	100				
		2	$r,\quad u_{1:2},\quad \beta_{1:3}$	6	120				
		3	$r,\quad u_{1:3},\quad \beta_{1:3}$	7	140				
#5	ASGD	1	$r,\quad u_1,\quad \alpha_{1:3},\quad p$	6	120	$o, l, r, u_{1:2}, \alpha_{1:3}, \beta_{1:3}, p$	13	8	680
		2	$r,\quad u_{1:2},\quad \alpha_{1:3},\quad p$	7	140				
	ADAM	1	$r,\quad u_1,\quad \beta_{1:3},\quad p$	6	120				
		2	$r,\quad u_{1:2},\quad \beta_{1:3},\quad p$	7	140				
		3	$r,\quad u_{1:3},\quad \beta_{1:3},\quad p$	8	160				

Table 1: Problem instances of the MLP example and their dataset sizes, with $u_{1:2} = (u_1, u_2)$, $u_{1:3} = (u_1, u_2, u_3)$, $\alpha_{1:3} = (\alpha_1, \alpha_2, \alpha_3)$ and $\beta_{1:3} = (\beta_1, \beta_2, \beta_3)$; the activation function is fixed at $a = \texttt{ReLU}$ for all instances

For Sub, the dataset size of a subproblem is 20 times the number of variables. For Graph, the dataset size is the sum of the subproblem dataset sizes. For both approaches, 50% of a dataset is allocated to a training set, and the validation and testing sets are each allocated a 25%. The data is generated with a uniform distribution on the restricted sets. Instance #1 fixes the optimizer, hence the number of hidden layers is meta instead of meta-decreed. There are 3 subproblems, each assigned to a fixed number of hidden layers. Instance #2 adds the hyperparameters $\alpha_{1:3}$ to the first instance. Instance #3 frees the optimizer $o \in \{\texttt{ASGD}, \texttt{ADAM}\}$, and the hyperparameters $\beta_{1:3}$ are introduced via $o = \texttt{ADAM}$. There are 4 subproblems, each assigned to a pair (o, l). Instance #4 adds a subproblem by allowing $l = 3$ when $o = \texttt{ADAM}$. The number of hidden layers l becomes a meta-decreed variable, since $L_{\texttt{ASGD}} \neq L_{\texttt{ADAM}}$. Instance #5 adds the dropout ρ with bounds in (1).

5.1 Setup

The generation of a data couple $(x, f(x))$ is done by the following steps: 1) a MLP model is constructed with respect to its given hyperparameters $x \in \mathcal{X}$ with PyTorch; 2) the MLP is trained and validated on the quarter of the Fashion-MNIST training dataset (15k data), and 3) the performance score $f(x) \in [0, 100]$, that represents the percentage of well-classified images, is computed on the MLP with the quarter of the Fashion-MNIST test dataset (2.5k data). To reduce data generation time, only 25% of the Fashion-MNIST dataset is used, and the number of epochs and batch size are respectively set to 25 and 128. Each instance has a heterogeneous dataset that is partitioned into a training, validation and test. For Sub, these datasets are further divided into subproblems according to the size column in Table 1.

Two types of regression models are considered, the inverse distance weighting (IDW) and the K-nearest neighbors (KNN). For IDW, the training dataset is composed of data points and images that are used for interpolation. For KNN, the training dataset contains data points that are available as neighbors, and the images of these neighbors are used to compute a mean. Graph constructs a single model with an induced distance. Sub constructs many models, one per subproblem, and each model utilizes an Euclidean distance defined on their corresponding subdomain.

In the computational experiments, a prediction error is computed with a Root Mean Squared Error (RMSE) on a test dataset. For Sub, the computation of the RMSE test is done by assigning each test data point to its respective subproblem and model. Before computing the RSME test, parameters are adjusted with respect to the RMSE on the validation dataset. The parameters to adjust depend on the model type and the approach employed. Regarding the model type, IDW adds no parameter, whereas KNN adds the number of neighbors K for Graph and a number of neighbors per subproblem for Sub. Both approaches have weight parameters. Graph also requires parameters for the excluded-included distances between variables that can take the special value EXC, and a parameter for the categorical distance of the optimizer (if not fixed). The optimization of the RMSE validation is done with the open-source blackbox optimization software NOMAD [8] that is based on the Mesh Adaptive Direct Search algorithm [6] (MADS).

5.2 Results

Table 2 details the results of RMSE validations and tests, and it presents the budget of evaluations allocated to the adjustment of parameters with NOMAD. Graph has fewer parameters, since all the variables are considered simultaneously and the number of bounds θ_i^r required is small for the instances.

In Table 2, the RMSE validation value is the smallest value during its optimization. The smallest value between Sub and Graph appear in bold. For the RMSE validations, Sub performs slightly better than Graph on all instances with the IDW model, whereas the results are mixed between the two approaches for the KNN model. For the RMSE tests, Graph outperforms Sub with both model types, except for the instance #5 with the IDW model. The best results are obtained using the KNN model with Graph. Graph seems generally more robust to overfitting as the gaps between its RMSE validations and tests are smaller: this is not surprising, since this approach uses all the data across the subproblems, and should therefore yield better generalization. Graph requires more time to generate the results in Table 2, since the interpolation (IDW) or the selection of neighbors (KNN) is done on all points. The worst gap between the two approaches is for the instance #5 with the IDW model type: 5 vs 30 minutes of CPU time to run the 3,500 budget of evaluations. The times were estimated with a 11th generation Intel i7-11800H (2.30 GHz) CPU.

		Budget	# of eval.		# of param.		RMSE validation		RMSE test	
Instance	Model	of eval.	Sub	Graph	Sub	Graph	Sub	Graph	Sub	Graph
#1	IDW	1,500	1,500	1,500	9	7	**11.49**	**11.49**	**13.12**	**13.12**
	KNN		1,447	1,078	12	9	**6.59**	6.97	10.90	**9.7**
#2	IDW	2,000	2,000	2,000	18	10	**16.24**	16.4	21.87	**21.31**
	KNN		2,000	1,931	21	11	12.52	**8.48**	21.40	**15.00**
#3	IDW	2,500	2,500	2,500	22	19	**12.59**	13.36	19.58	**19.13**
	KNN		2459	2,500	26	20	9.31	**8.56**	18.83	**15.15**
#4	IDW	3,000	3,000	3,000	29	21	**12.14**	12.69	18.58	**18.39**
	KNN		3,000	2,219	34	22	9.64	**8.24**	15.71	**14.93**
#5	IDW	3,500	3,500	3,500	34	22	**13.19**	14.42	**16.5**	16.94
	KNN		3,500	2,404	39	23	**10.12**	11.28	17.55	**14.29**

Table 2: Number of parameters, RMSE validation datasets and RMSE datasets on the problem instances.

5.2.1 Experiments on training dataset with selected parameters

In this section, some computational experiments are done to further study the aggregation of data. Graphs plot the RMSE test versus the number of points in a (partial) training dataset. Models are constructed with a partial training dataset, which is increased iteratively with an additional random data point drawn from the training dataset without replacement. The partial training dataset begins with a random data point from each subproblem, and points are added until it becomes the training dataset. The adjusted parameters obtained from previous Section 5.2 are selected and fixed for all iterations.

In Figures 4, the solid lines represents means, and the shaded areas represent standard deviations. As the level of difficulty increases through the instances, Graph performs progressively better than Sub for the IDW model, except for the last iterations in instance #5. Graph outperforms Sub on all instances with the KNN model, and particularly for the instance #1. The aggregation of data seems particularly promising for the KNN model as it provides access to more neighbors across the subproblems.

5.2.2 Experiments on training dataset with optimized parameters

A final experiment is conducted on instance #1, which provided the worst relative performance of Graph with the IDW model. The experiment is similar to the ones in previous Section 5.2.1, except that the parameters are now optimized with respect to RMSE validation at each iteration, and points are added in the partial training dataset until it reaches 30 points. Each iteration has a budget of 250 evaluations for the parameters adjustment with the software NOMAD.

Figure 5 is composed of 10 runs of different random seeds. Graph performs considerably better and its standard deviation (shaded) is globally smaller. This experiment is more realistic than the previous ones, since parameters are not fixed and provided in advanced. In comparison to the experiment with selected parameters, this experiment indicates that aggregating training data points is particularly beneficial for obtaining lower RMSE test.

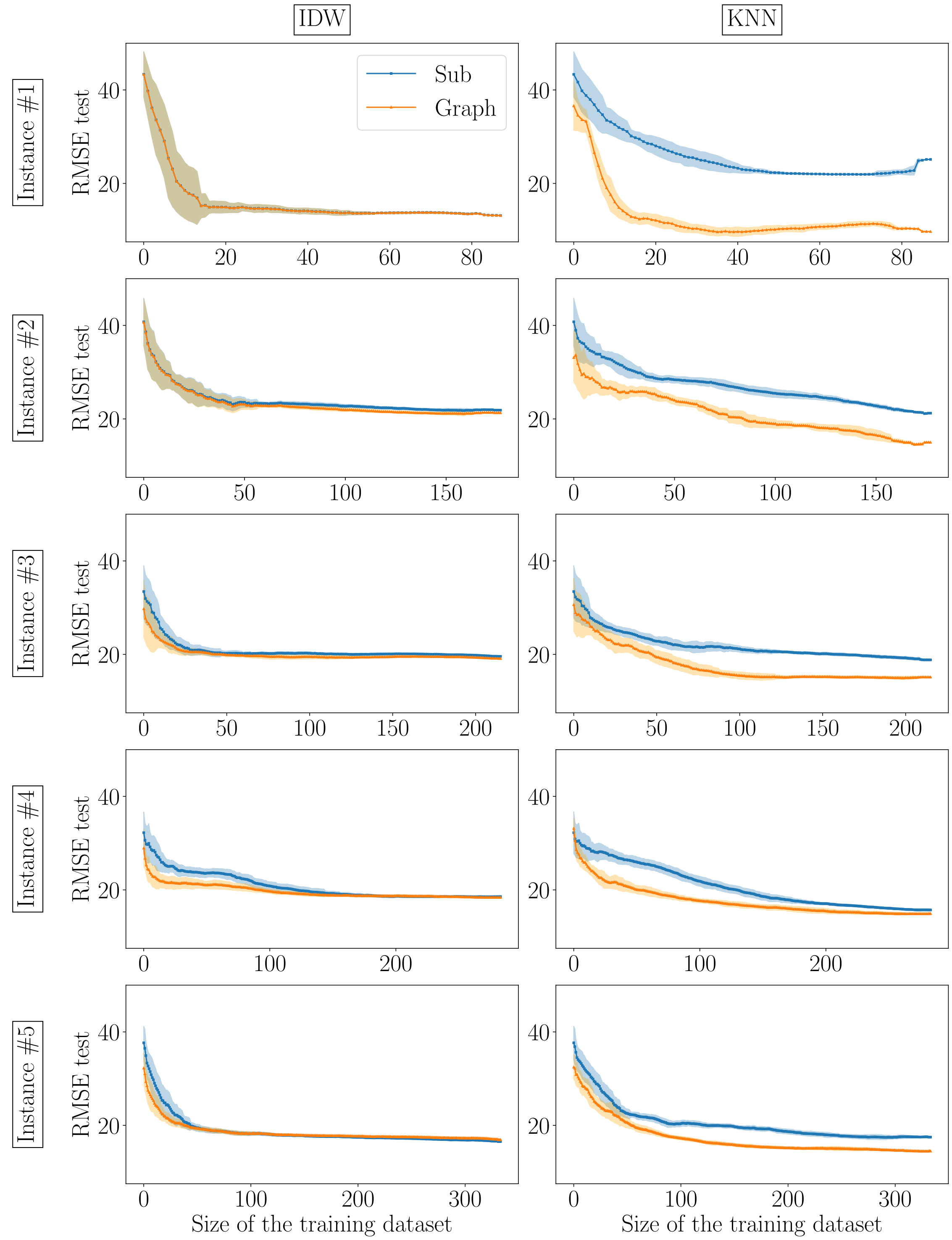

Figure 4: RMSE tests with iteratively increasing training data points and selected parameters. Each subfigure is composed of 30 runs of different random seeds. The same vertical scale is used for comparability.

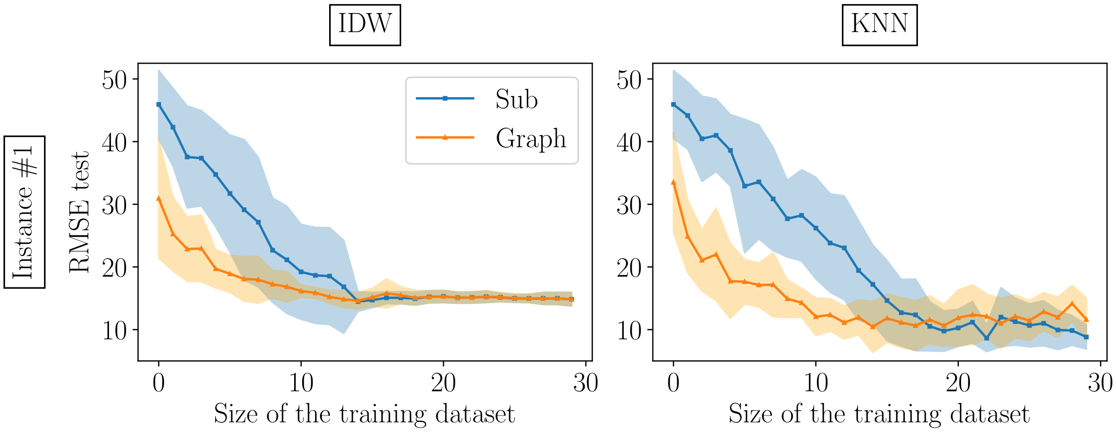

Figure 5: RMSE tests with iteratively increasing training data points and optimized parameters.

6 Discussion

The present work focused on heterogeneous datasets that are intrinsically related to mixed-variable domains with meta and meta-decreed variables. The first important contribution, introduced in Section 3, is a generalized modeling framework for such domains, called graph-structured. The modeling framework is rigorously constructed through graph theory, and it introduces many definitions and structures, such as meta-decreed variables, the role graph $G = (V, A)$, excluded variables, extended points and the extended domain. As such, the paper generalizes the following state-of-the-art frameworks: mixed-variable domains with (strictly) meta variables [7]; tree-structured spaces [12]; hierarchical spaces [20]; and variable-size design space [28]. The second important contribution, detailed in Section 4, concerns distance functions, which allows computation of distances between mixed-variable points that do not share the same variables. The graph-structured distance considers both the included and excluded variables in an extended point, and is constructed variable-wise with one-dimensional included-excluded to facilitate computations. Computational experiments on the MLP example are done in Section 5. The graph-structured approach aggregates data across subproblems, and outperforms the approach that separates a problem into subproblems. Further computational experiments on diverse problems and models are required to confirm that the modeling framework and the graph-structured distance are state-of-the-art for heterogeneous datasets. For instance, Gaussian Processes are commonly used for such problems [28, 33, 34], and they will be studied in next work. In the near future, time consumption will also be studied more thoroughly with more complex models.

Data availability statement

Scripts and data are publicly available at https://github.com/bbopt/graph_distance.

Conflict of interest statement

On behalf of all authors, the corresponding author states that there is no conflict of interest.

Acknowledgments

We express our gratitude to Amaury Diopus'kin for the PyTorch implementation, produced during its internship in summer 2022, that was used for data generation.

References

[1] A. Ahmad and L. Dey. A k-mean clustering algorithm for mixed numeric and categorical data. *Data & Knowledge Engineering*, 63(2):503–527, 2007.

[2] A. Aleti., B. Buhnova, L. Grunske, A. Koziolek, and I. Meedeniya. Software Architecture Optimization Methods: A Systematic Literature Review. *IEEE Transactions on Software Engineering*, 39(5):658–683, 2013.

[3] N. Ali, D. Neagu, and P. Trundle. Classification of Heterogeneous Data Based on Data Type Impact on Similarity. In *Advances in Computational Intelligence Systems*. Springer International Publishing, 2019.

[4] N. Ali, D. Neagu, and P. Trundle. Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets. *SN Applied Sciences*, 1:1–15, 2019.

[5] M. Asadi, G. Gröner, B. Mohabbati, and D. Gašević. Goal-oriented modeling and verification of feature-oriented product lines. *Software & Systems Modeling*, 15:257–279, 2016.

[6] C. Audet and J.E. Dennis, Jr. Mesh Adaptive Direct Search Algorithms for Constrained Optimization. *SIAM Journal on Optimization*, 17(1):188–217, 2006.

[7] C. Audet, E. Hallé-Hannan, and S. Le Digabel. A General Mathematical Framework for Constrained Mixed-variable Blackbox Optimization Problems with Meta and Categorical Variables. *Operations Research Forum*, 4(12), 2023.

[8] C. Audet, S. Le Digabel, V. Rochon Montplaisir, and C. Tribes. Algorithm 1027: NOMAD version 4: Nonlinear optimization with the MADS algorithm. *ACM Transactions on Mathematical Software*, 48(3):35:1–35:22, 2022.

[9] D. Batory. Feature models, grammars, and propositional formulas. In *International Conference on Software Product Lines*. Springer, 2005.

[10] G. Beer. The Structure of Extended Real-valued Metric Spaces. *Set-Valued and Variational Analysis*, 21:591–602, 2013.

[11] D. Benavides, S. Segura, and A. Ruiz-Cortés. Automated analysis of feature models 20 years later: A literature review. *Information Systems*, 35(6):615–636, 2010.

[12] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems*, 2011.

[13] J.H. Bussemaker, P.D. Ciampa, T. De Smedet, B. Nagel, and G. La Rocca. System Architecture Optimization: An Open Source Multidisciplinary Aircraft Jet Engine Architecting Problem. In *AIAA AVIATION 2021 Forum*, AIAA AVIATION Forum. American Institute of Aeronautics and Astronautics, 2021.

[14] J. Choo, S. Bohn, G.C. Nakamura, A.M. White, and H. Park. Heterogeneous Data Fusion via Space Alignment Using Nonmetric Multidimensional Scaling. In *SIAM International Conference on Data Mining*. SIAM, 2012.

[15] M. Feurer and F. Hutter. Hyperparameter optimization. *Automated machine learning: Methods, systems, challenges*, pages 3–33, 2019.

[16] A. Gardner, C.A. Duncan, J. Kanno, and R.R. Selmic. On the Definiteness of Earth Mover's Distance and Its Relation to Set Intersection. *IEEE Transaction on Cybernetics*, 48(11):3184–3196, 2018.

[17] E.C. Garrido-Merchán and D. Hernández-Lobato. Dealing with categorical and integer-valued variables in Bayesian Optimization with Gaussian processes. *Neurocomputing*, 380:20–35, 2020.

[18] K. Horio, S. Ishikawa, and R. Kubota. Effective Hierarchical Optimization using Integration of Solution Spaces and its Application to multiple Vehicle Routing Problem. In *2015 International Symposium on Intelligent Signal Processing and Communication Systems*. IEEE, 2015.

[19] L.-Y. Hu, M.-W. Huang, S.-W. Ke, and C.-F. Tsai. The distance function effect on k-nearest neighbor classification for medical datasets. *SpringerPlus*, 5:1–9, 2016.

[20] F. Hutter and M.A. Osborne. A Kernel for Hierarchical Parameter Spaces. Technical Report 1310.5738, ArXiv, 2013.

[21] R. Jin and H. Liu. A Novel Approach to Model Generation for Heterogeneous Data Classification. In *IJCAI International Joint Conference on Artificial Intelligence*, Proceedings of the 19th international joint conference on Artificial intelligence. Morgan Kaufmann Publishers Inc., 2005.

[22] K. Kim and J. s. Hong. A hybrid decision tree algorithm for mixed numeric and categorical data in regression analysis. *Pattern Recognition Letters*, 98:39–45, 2017.

[23] D. Lakhmiri, S. Le Digabel, and C. Tribes. HyperNOMAD: Hyperparameter Optimization of Deep Neural Networks Using Mesh Adaptive Direct Search. *ACM Transactions on Mathematical Software*, 47(3), 2021.

[24] S. Lucidi and V. Piccialli. A Derivative-Based Algorithm for a Particular Class of Mixed Variable Optimization Problems. *Optimization Methods and Software*, 17(3–4):317–387, 2004.

[25] S. Lucidi, V. Piccialli, and M. Sciandrone. An Algorithm Model for Mixed Variable Programming. *SIAM Journal on Optimization*, 15(4):1057–1084, 2005.

[26] G. Noroozi. Data Heterogeneity and Its Implications for Fairness. Master's thesis, Western University, 2023. Available at https://ir.lib.uwo.ca/etd/9623/.

[27] J. Pelamatti, L. Brevault, M. Balesdent, E.-G. Talbi, and Y. Guerin. Efficient global optimization of constrained mixed variable problems. *Journal of Global Optimization*, 73(3):583–613, 2019.

[28] J. Pelamatti, L. Brevault, M. Balesdent, E.-G. Talbi, and Y. Guerin. Bayesian optimization of variable-size design space problems. *Optimization and Engineering*, 22:387–447, 2021.

[29] C.L. Pereira, G.D.C. Cavalcanti, and T.I. Ren. A New Heterogeneous Dissimilarity Measure for Data Classification. In *IEEE International Conference on Tools with Artificial Intelligence*. IEEE, 2010.

[30] P.Z.G. Qiand, H. Wu, and C.F.J. Wu. Gaussian process models for computer experiments with qualitative and quantitative factors. *Technometrics*, 50(3):383–396, 2008.

[31] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[32] P. Saves. *High-dimensional multidisciplinary design optimization for aircraft eco-design*. PhD thesis, ONERA and ISAE-SUPAERO, 2024. Available at https://theses.hal.science/ONERA-MIP/tel-04439128v1.

[33] P. Saves, Y. Diouane, N. Bartoli, T. Lefebvre, and J. Morlier. A mixed-categorical correlation kernel for Gaussian process. *Neurocomputing*, 550:126472, 2023.

[34] P. Saves, R. Lafage, N. Bartoli, Y. Diouane, J. H. Bussemaker, T. Lefebvre, J. T. Hwang, J. Morlier, and J.R.R.A Martins. SMT 2.0: A Surrogate Modeling Toolbox with a focus on Hierarchical and Mixed Variables Gaussian Processes. *Advances in Engineering Software*, 188:103571, 2024.

[35] Y.-Y. Song and Y. Lu. Decision tree methods: applications for classification and prediction. *Shanghai Arch Psychiatry*, 27(2):130–135, 2015.

[36] D. J. Wild. Mining large heterogeneous data sets in drug discovery. *Expert Opinion on Drug Discovery*, 4(10):995–1004, 2009.

[37] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H.L., and S.-H. Deng. Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization. *Journal of Electronic Science and Technology*, 17(1):26–40, 2019.

[38] L. Yang and A. Shami. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415:295–316, 2020.

[39] L. Zhang, Y. Xie, L. Xidao, and X. Zhang. Multi-source heterogeneous data fusion. In *International Conference on Artificial Intelligence and Big Data (ICAIBD)*. IEEE, 2018.

[40] Y. Zhang, S. Tao, W. Chen, and D.W. Apley. A Latent Variable Approach to Gaussian Process Modeling with Qualitative and Quantitative Factors. *Technometrics*, 62(3):291–302, 2020.