Heterogeneous Metric Learning of Categorical Data with Hierarchical Couplings

Chengzhang Zhu, Longbing Cao, *Senior Member, IEEE,* Qiang Liu, *Member, IEEE*
Jianping Yin and Vipin Kumar, *Fellow, IEEE*

Abstract—Learning appropriate metric is critical for effectively capturing complex data characteristics. The metric learning of categorical data with hierarchical coupling relationships and local heterogeneous distributions is very challenging yet rarely explored. This paper proposes a Heterogeneous mEtric Learning with hIerarchical Couplings, HELIC, for this type of categorical data. HELIC captures both low-level value-to-attribute and high-level attribute-to-class hierarchical couplings, and reveals the intrinsic heterogeneities embedded in each level of couplings. Theoretical analyses of the effectiveness and generalization error bound verify that HELIC effectively represents the above complexities. Extensive experiments on 30 data sets with diverse characteristics demonstrate that HELIC-enabled classification significantly enhances the accuracy (up to 40.93%), compared with five state-of-the-art baselines.

Index Terms—Metric Learning, Distance Metric, Similarity Measure, Coupling Learning, Heterogeneity Learning, non-IID Learning, Categorical Data

1 Introduction

Developing distance metrics that can effectively capture complex data characteristics in categorical data is a fundamental yet challenging task, which largely determines the learning quality. Although considerable work has been conducted to measure numerical data, only limited efforts have been made on categorical data. As discussed in [1] etc., it is not as straightforward to measure categorical data similarity as it to measure numerical distance. This is due to the significantly greater challenges in understanding and representing categorical data complexities: (1) Categorical values are nominal, i.e., without explicit intervals (note: in this paper we do not consider ordinal categorical data) and (2) there are hierarchical couplings and heterogeneities that essentially determine data complexities.

Target Problems & Gap Analysis This paper thus focuses on learning the distance metrics for categorical data embedded with sophisticated coupling relationships (couplings for short) and heterogeneities. Here *couplings* refer to the interactions within and between attributes, objects, and classes; and *heterogeneities* relate to different parts of data that hold diverse relationships and distributions [2], [3], [4].

Most of the existing metric learning methods handle numerical data [5], [6], [7], [8], [9], [10], [11], [12], [13]. Although these methods can learn the distance in numerical data, they cannot handle categorical data directly. While categorical input is involved in work such as [14], [15], [16], they ignore the above-discussed couplings and heterogeneities.

In recent years, several distance metrics or measures have been proposed to capture intra- and inter-attribute couplings in categorical data. For example, the conditional probability [17] and rough membership function [18] capture intra-attribute couplings. The inter-attribute conditional probability in [1], [19], [20] and the co-occurrence frequency of highly interdependent attributes in [21] measure inter-attribute couplings. A novel categorical data distance measure named *coupled object similarity* (COS) [22], [23] learns and integrates the intra- and inter-attribute couplings.

However, the above methods for learning feature interactions treat the distributions for value, attribute and object in categorical data as homogeneous. They usually adopt one measure for all data but ignore the difference between values (and attributes and objects) and their relationships. Hence, these methods cannot represent the heterogeneity in categorical data. In [24], multiple distances are learned to handle more than one type of categorical relationship, but this method ignores the hierarchical interactions in categorical data.

As a result of overlooking or insufficiently representing couplings and/or heterogeneities in metric learning, the learned metric is ineffective for approximating data complexities and measuring distances in a categorical space.

Our Design & Main Contributions

To tackle the above issues, this paper introduces a novel data-driven Heterogeneous mEtric Learning of hIerarchical Couplings (HELIC) for representing categorical data. First, HELIC captures both low-level value-to-attribute and high-level attribute-to-class couplings to comprehensively reveal the intrinsic and hierarchical characteristics in categorical data. HELIC captures the following interactions: (1) the relationships between the values of an attribute, called *intra-attribute couplings*, to measure the within-attribute similarities. Such couplings reflect the value interactions within an attribute; (2) the relationships between attributes, called *inter-attribute couplings*, to measure the between-attribute similarities. These couplings describe the interactions between attribute values conditional on other attributes; and (3) the relationships between attributes and classes, called *attribute-class couplings*, to measure the attribute-class similarities. These couplings reveal the value distribution w.r.t. each class. Second, HELIC reveals

Chengzhang Zhu and Longbing Cao are with the Advanced Analytics Institute, University of Technology Sydney, Australia. E-mail: kevin.zhu.china@gmail.com, Longbing.Cao@uts.edu.au. Qiang Liu is with the College of Computer, National University of Defense Technology, China. Jianping Yin is with the Dongguan University of Technology, China. Vipin Kumar is with the Department of Computer Science, University of Minnesota, USA.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2791525, IEEE Transactions on Knowledge and Data Engineering

the intrinsic heterogeneities across various types of couplings to identify their different local structures and distributions. Lastly, HELIC learns a heterogeneous metric based on the captured couplings and heterogeneity.

The key contributions of this work include:

- *Learning hierarchical couplings*: several learning functions are proposed to represent the hierarchical couplings in categorical data, i.e., the value-to-class (including value-to-attribute and attribute-to-class) couplings within and between categorical attributes and between attributes and classes. These data-driven hierarchical value-to-class couplings complement and enhance metric performance.
- *Learning heterogeneities*: a learning model is proposed to learn and integrate heterogeneous local relationships in categorical data. The hierarchical value-to-class couplings are incorporated into different kernel functions, and a transformed matrix and combination coefficient are learned for each kernel to reveal the corresponding distributions of specific values and class labels. The learned heterogeneous information represents the different local views of categorical data relationships.
- *Theoretical analysis of effectiveness*: we prove the effectiveness of HELIC in terms of improving metric learning accuracy from a theoretical aspect. The learning generalization error bound is also analyzed. The theoretical results explain why HELIC effectively discloses the complex relationships in categorical data.

We compare HELIC with five state-of-the-art distance measures on 30 data sets with different data characteristics. The experimental results show that HELIC significantly improves the learning performance.

The rest of this paper is organized as follows. Section 2 discusses the related work and gaps. Section 3 outlines the HELIC framework and introduces its design. Section 4 presents the theoretical analysis of the HELIC properties. Section 5 demonstrates the HELIC performance by comparing it with existing categorical distance measures from a variety of aspects. Lastly, Section 6 concludes this paper.

2 Related Work

Distance metrics significantly affect learning performance. Typically, a poorly-designed distance metric cannot induce good learning performance. On the contrary, a discriminative one is more likely to enable better learning outcomes.

Typical categorical data distance metrics are matching- and frequency-based. Compared with matching-based measurements, e.g., Hamming distance, frequency-based methods can capture more information. For example, the co-occurrence frequencies of attribute values in samples is measured in [25], which reflects the distance between the same nominal value in different objects but does not capture the additional information on different values. In [26], an occurrence frequency-based method evaluates the distance between different values. Although these methods reveal distance information in some aspects, they do not capture the hierarchical coupling relationships [27], including the within and between attribute interactions and the interactions between attributes and labels, which fundamentally determine object distances.

Recent efforts in categorical metric learning have been made on capturing various couplings. The work in [1], [17], [18], [19], [20], [21] proposes more sophisticated tools to measure the distance between different values according to value frequencies. According to the captured information, they can be divided into two groups: (1) intra-attribute information-based measures [17], [18]; and (2) inter-attribute information-based measures [1], [19], [20], [21].

In the first group, the method in [17] adopts the conditional probability of an attribute value of an object with respect to a cluster center to calculate the distance between an object and the cluster center. However, this method has to update the measured similarity after every clustering step, making it sensitive to clustering methods and inefficient. Inspired by biological and genetic taxonomy, a rough set-based membership function measures the distance of attribute values inspired by biological and genetic taxonomy in [18]. Although it is claimed to reveal more information than the simple matching method, its main weakness lies in that the rough set theory only provides detailed similarity information of the same attribute values. The above methods measure categorical value distance by only using intra-attribute information, but ignore the inter-attribute information.

The second group involves inter-attribute information. For example, in [19], the distance of categorical values is measured w.r.t. their relationships with other attributes. Given two values of an attribute, it considers the distance between the conditional probability of other attributes. Specifically, it first calculates the conditional probabilities of a value of another attribute in terms of the given values. Then, it feeds the two conditional probabilities into the Kullack-Leibler divergence method [28] to measure the distance of given values. Following the work in [19], another inter-attribute information-based metric [20] uses the maximum probability of two values' divergence in another attribute to calculate their distance. Both methods assume each attribute affects other attributes significantly, which may not always hold [3]. To tackle this problem, the Symmetric Uncertainty (SU) in [1] measures the correlation of two attributes, but it only considers the inter-attribute information between attributes with strong relevance but small redundancy. In [21], the co-occurrence frequencies of highly inter-dependent attributes are included into the measurement. However, the above methods consider intra-attribute information and inter-attribute information separately. To involve both intra- and inter-attribute couplings and their combinations in measuring object similarity, the similarity measure, coupled object similarity (COS) in [22], [23], is the first method to capture complex hierarchical couplings in categorical data. However, COS assumes homogeneous data distributions, which does not hold in real-life data.

In addition, another set of methods, e.g., [14], [16], [29], [30], [31], [32], [33], learns metrics by using side-information such as labels, to guide metric learning for categorical data. The work in [29] is the first to consider label information for categorical data similarity, which uses labels to divide data into subsets and considers the attribute value distribution within these subsets. The difference in the attribute values' appearance frequency is used as a distance measure, called the Value Difference Metric (VDM). However, VDM only considers the categorical value distribution in a label-induced subset and treats each attribute separately, which may cause overfitting. The method in [14] revises VDM by considering inter-attribute information to learn a similarity measure, achieving better classification performance but being time consuming. Instead of calculating the metric directly, the work in [30] learns a vector representation for a categorical value guided by the label information. It optimizes vector representation to guarantee that the intra-class distance is smaller than the inter-class distance. Another method in [31] studies the connection between metric learning and kernel learning by transforming a metric learning problem to a kernel learning task so that the similarity of different types of data, including categorical data, can be measured. Although these two methods can learn metrics for categorical data, they ignore data characteristics and have high computational complexity. The work in [32] proposes a kernel density metric learning (KDML) method with a non-linear and probability-based similarity measure. However, KDML suffers from information loss due to only using the matching method to capture the relationships in the data. Instead of considering class as side information, the method in [33] captures the class information and classification model information simultaneously. However, it only maps a categorical value to a numerical value, which cannot well represent a categorical value when it has high-dimensional embedding. Multiple distances are learned in [24] to leverage the heterogeneous categorical relationships, while the hierarchical couplings are ignored. In [16], a metric learning method fits the ordinal data characteristic but is not suitable for learning metrics for general categorical data owing to its strong ordinal assumption.

In addition, there are many methods for learning distance metrics on numerical data [5], [6], [7], [9], [12], [13], [34]. They cannot handle categorical data directly or be simply converted to measure categorical data because: (1) they usually represent data w.r.t. numerical vectors with certain intervals, which violates the nature of categorical data; (2) transforming categorical data to an appropriate numerical representation is nontrivial, as shown in [35].

In this work, the proposed HELIC directly learns categorical data distance metrics. It considers the hierarchical couplings from value to attribute, object and class and measures their respective distances in terms of frequency and co-occurrence in categorical data. Further, HELIC learns the heterogeneities in categorical data by incorporating hierarchical value-to-class couplings into different kernel functions to learn the corresponding distributions of respective entities.

3 THE HELIC DESIGN

In this section, we introduce the working mechanisms of HELIC and its components.

3.1 Problem Statement

Generally speaking, a categorical data set can be represented as a three-element tuple $S =< O, A, V >$, where $O = \{\mathsf{o}_i | i \in N_o\}$ is the object set with n_o elements o_i, and N_o is the set of index for objects; $A = \{\mathsf{a}_i | i \in N_a\}$ is the attribute set with n_a elements a_i; and $V = \bigcup_{j=1}^{n_a} V^{(j)}$ is the collection of attribute values with n_v elements, in which $V^{(j)} = \{\mathsf{v}_i^{(j)} | i \in N_v^{(j)}\}$ is the set of attribute values $\mathsf{v}_i^{(j)}$ with $n_v^{(j)}$ elements of attribute a_j. For the above tuple, $v_i^{(j)}$ is the value of the i-th object in j-th attribute. In supervised cases, a class set $C = \{\mathsf{c}_i | i \in N_c\}$ partitions data into n_c classes. For the i-th object, c_i refers to its class. Table 1 lists the symbol styles used in this paper.

The metric learning for categorical data aims to learn a distance metric $d(\cdot, \cdot) : \mathcal{O} \times \mathcal{O} \to \mathcal{R}_0^+$ for all categorical objects in a data set O that satisfies the properties:

1) $d(\mathsf{o}_i, \mathsf{o}_j) + d(\mathsf{o}_j, \mathsf{o}_k) \geq d(\mathsf{o}_i, \mathsf{o}_k)$,
2) $d(\mathsf{o}_i, \mathsf{o}_j) \geq 0$,
3) $d(\mathsf{o}_i, \mathsf{o}_j) = d(\mathsf{o}_j, \mathsf{o}_i)$.

TABLE 1
List of Symbols

Symbol	Symbol Style
element	lowercase with sans serif font
value	lowercase
vector	lowercase with bold font
matrix	uppercase with bold font
set	uppercase
function	lowercase with parentheses
space	uppercase with calligraphic font
value index	subscript
attribute index	superscript with parenthesis

Denoting the representation of objects in the learned metric space as $\mathbf{x}$ that $d(\mathsf{o}_i, \mathsf{o}_j) = \mathbf{x}_i \odot \mathbf{x}_j$, where $\odot$ refers to any kinds of operations between two vectors. The learned distance metric should also minimize the divergence between the data distribution in the categorical space $\mathfrak{O}$ and the data distribution in the metric space $\mathfrak{X}$. Although it is hard to measure the divergence directly, the divergence can be approximated through involving some side information (like class label) for specific problems. Given an approximate divergence measure $\widetilde{Div}(\cdot||\cdot)$ based on side information, the objective function of metric learning for categorical data can be formalized as follows,

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \widetilde{Div}(\mathfrak{O}||\mathfrak{X}) \\ \text{subject to} \quad & \mathsf{o} \sim \mathfrak{O} \\ & \mathbf{x} \sim \mathfrak{X} \\ & d(\mathsf{o}_i, \mathsf{o}_j) = \mathbf{x}_i \odot \mathbf{x}_j . \end{aligned} \tag{1}$$

In this paper, the proposed HELIC learns a metric for categorical data. It satisfies all metric properties and simultaneously captures the hierarchical couplings and heterogeneity in categorical data to minimize the divergence between data distributions in the categorical space and the metric space.

3.2 The HELIC Framework

Fig. 1 illustrates the framework of HELIC, which has a three-layer hierarchical structure for coupling learning, heterogeneity learning, and metric learning. At the coupling learning stage, HELIC maps categorical data into three coupling spaces: intra-attribute couplings, inter-attribute couplings, and attribute-class couplings, to reveal the value-to-class couplings. At the heterogeneity learning stage, HELIC models each type of couplings in the coupling spaces by specific kernel functions, and learns a transformed matrix for each kernel to construct a heterogeneous kernel space. As a result, the various relationships corresponding to respective kernels are revealed. Lastly, at the metric learning stage, a distance metric is learned from the heterogeneous kernel spaces by involving a hypothesis and/or side information (e.g., label, ordinal and semantic information) relevant to a learning task.

3.3 Basic Information Functions

Let the relationships between objects, attributes and attribute values be represented by a set of functions $\{v^{(j)}(\cdot) | j \in N_a\}$, in which $v^{(j)}(\cdot) : O \to V^{(j)}$ maps an object to a particular value w.r.t. attribute a_j. Let the relationships between objects and

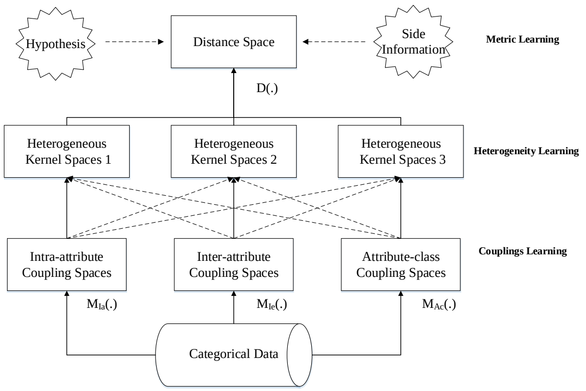

Hypothesis
Distance Space
Side Information
Metric Learning
D(.)
Heterogeneous Kernel Spaces 1
Heterogeneous Kernel Spaces 2
Heterogeneous Kernel Spaces 3
Heterogeneity Learning
Intra-attribute Coupling Spaces
Inter-attribute Coupling Spaces
Attribute-class Coupling Spaces
Couplings Learning
M_Ia(.)
M_Ie(.)
M_Ac(.)
Categorical Data

Fig. 1. The HELIC Framework: The coupling learning first represents the couplings in the categorical data, then the heterogeneity learning reveals the heterogeneous distributions of categorical values in the coupling spaces and feeds them into metric learning.

classes be represented by the function $c(\cdot) : O \rightarrow C$, which maps an object to its corresponding class or classes. For example, in Table 2, a relationship between object, attribute and value for A2 is $v^{(2)}(\text{A2}) = yellow$, and a relationship between object and class for A2 is $c(\text{A2}) = low$. With this, we define the following basic information functions to learn the hierarchical couplings in our proposed methods.

TABLE 2
Toy Example: The watermelon information table

ID	Texture	Color	Root Shape	Sweetness
A1	clear	white	straight	low
A2	blurry	yellow	straight	low
A3	blurry	yellow	curled	low
A4	clear	green	slightly curled	low
A5	blurry	green	curled	high
A6	clear	black	slightly curled	high

Definition 1 (Attribute-to-Object Mapping Function (AOF))**.** AOF, denoted as $g^{(j)}(\cdot)$, is a mapping function that maps values in a value set of the j-th attribute to their corresponding objects in the data set.

$$g^{(j)}(V_*^{(j)}) = \{\mathsf{o}_i | v^{(j)}(\mathsf{o}_i) \in V_*^{(j)}\}, \tag{2}$$

where $V_*^{(j)} \subseteq V^{(j)}$ is a subset of the attribute $\mathsf{a}^{(j)}$'s value set. Accordingly, $g^{(j)}(V_*^{(j)})$ returns those objects having the given values in $V_*^{(j)}$.

For example, in Table 2, $g^{(3)}(\{slightly\, curled\})$ = {A4, A6} and $g^{(2)}(\{white, black\})$ = {A1, A6}.

Definition 2 (Class-to-Object Mapping Function (COF))**.** COF, denoted as $h(\cdot)$, is a mapping function that finds those objects whose class labels are contained in a given class set.

$$h(C_*) = \{\mathsf{o}_i | c(\mathsf{o}_i) \in C_*\}, \tag{3}$$

where $C_* \subseteq C$ is a subset of all classes.

For example, $h(\{low\})$ = {A1, A2, A3, A4} and $h(\{high\})$ = {A5, A6} as shown in Table 2.

Definition 3 (Information Conditional Probability Functions (ICPF))**.** ICPF are functions to calculate the information conditional probability of the value set of a categorical attribute w.r.t. the set of another attribute, following Bayes' Theorem. To calculate two sets of values from different attributes, we denote ICPF as $p_{j|k}(\cdot|\cdot)$. To calculate a set of values and a set of classes, we denote ICPF as $p_{j|c}(\cdot|\cdot)$. Given value subset $V_*^{(j)}$ of attribute a_j and value subset $V_*^{(k)}$ of attribute a_k, ICPF $p_{j|k}(V_*^{(j)}|V_*^{(k)})$ is calculated as:

$$p_{j|k}(V_*^{(j)}|V_*^{(k)}) = \frac{|g^{(j)}(V_*^{(j)}) \cap g^{(k)}(V_*^{(k)})|}{|g^{(k)}(V_*^{(k)})|}. \tag{4}$$

Given value subset $V_*^{(j)}$ of attribute a_j and class subset C_*, the ICPF $p_{j|c}(V_*^{(j)}|C_*)$ is calculated as:

$$p_{j|c}(V_*^{(j)}|C_*) = \frac{|g^{(j)}(V_*^{(j)}) \cap h(C_*)|}{|h(C_*)|}. \tag{5}$$

In Eqs. (4) and (5), $\cap$ calculates the intersection of two sets, and $|\cdot|$ returns the number of elements in a given set.

For example, in Table 2, the ICPF of the value $curled$ of attribute *root shape* w.r.t. the value $yellow$ of attribute *color* is

$$p_{3|2}(\{curled\}|\{yellow\}) = \frac{|\{\text{A3, A5}\} \cap \{\text{A2, A3}\}|}{|\{\text{A2, A3}\}|} = \frac{1}{2},$$

and w.r.t. the class low is

$$\begin{aligned} & p_{3|c}(\{curled\}|\{low\}) \\ = & \frac{|\{\text{A3, A5}\} \cap \{\text{A1, A2, A3, A4}\}|}{|\{\text{A1, A2, A3, A4}\}|} \\ = & \frac{|\{\text{A3}\}|}{|\{\text{A1, A2, A3, A4}\}|} = \frac{1}{4}. \end{aligned}$$

For simplicity, in the following parts, we use $p(\cdot|\cdot)$ to represent the ICPF for both cases, i.e., between two sets of values and between a set of values and a set of classes, when it does not cause confusion.

3.4 Learning Value-to-Class Couplings

Learning Intra-attribute Couplings. *Intra-attribute couplings* represent the interactions between the values of an attribute. One way to observe such couplings is to analyze the value distributions in an attribute. We set a value frequency function to map the intra-attribute distributions to a numerical space, and the intra-attribute couplings are measured by the distance in this numerical space. For a categorical value $\mathsf{v}_i^{(j)}$ in the j-th attribute, the intra-attribute coupling learning function $m_{Ia}^{(j)}(\mathsf{v}_i^{(j)})$ maps an intra-attribute coupling between the value and other categorical values in this attribute to a one-dimensional vector,

$$m_{Ia}^{(j)}(\mathsf{v}_i^{(j)}) = \frac{|g^{(j)}(\mathsf{v}_i^{(j)})|}{n_o}. \tag{6}$$

An intra-attribute coupling space is spanned by the vector obtained in an attribute by Eq. (6) and is defined below:

$$\mathcal{M}_{Ia}^{(j)} = \{m_{Ia}^{(j)}(\mathsf{v}_i^{(j)}) | \mathsf{v}_i^{(j)} \in V^{(j)}\}. \tag{7}$$

For categorical data with n_a attributes, the intra-attribute coupling spaces are $\mathcal{M}_{Ia} = \{\mathcal{M}_{Ia}^{(1)}, \cdots, \mathcal{M}_{Ia}^{(n_a)}\}$.

An intra-attribute coupling space is a one-dimensional embedding of the categorical data space. Since a categorical value is often embedded in a high-dimensional attribute-value space, the intra-attribute coupling space cannot completely reflect the original attribute space. Inter-attribute couplings and attribute-attribute couplings are discussed in the following to capture the complementary information.

Learning Inter-attribute Couplings. *Inter-attribute couplings* refer to the interactions between attributes, which contain the contextual and/or semantic information of values w.r.t. other attributes. For the examples in Table 2, if the number of watermelons which are $white$ and $black$ in color are similar, but the values of the other attributes, e.g., their root shape, are significantly different, we can differentiate these $white$ and $black$ colored watermelons by including their root shape.

HELIC uses information conditional probability to present the inter-attribute couplings, which reveal the categorical value distributions in subspaces w.r.t. values in other attributes. For a categorical value $\mathsf{v}_i^{(j)}$ in the j-th attribute and the set of values in other attributes $V_* = \{V^{(k)}|k \in N_a, k \neq j\}$, the inter-attribute coupling learning function is formalized as follows:

$$m_{Ie}^{(j)}(\mathsf{v}_i^{(j)}) = [p(\mathsf{v}_i^{(j)}|\mathsf{v}_{*1}), \cdots, p(\mathsf{v}_i^{(j)}|\mathsf{v}_{*k}), \cdots, p(\mathsf{v}_i^{(j)}|\mathsf{v}_{*|V_*|})]^\top, \tag{8}$$

where $\mathsf{v}_{*l} \in V_*$. With this learning function, the inter-attribute coupling space is constructed as follows:

$$\mathcal{M}_{Ie}^{(j)} = \{m_{Ie}^{(j)}(\mathsf{v}_i^{(j)})|\mathsf{v}_i^{(j)} \in V^{(j)}\}. \tag{9}$$

For categorical data with n_a attributes, the inter-attribute coupling spaces $\mathcal{M}_{Ie} = \{\mathcal{M}_{Ie}^{(1)}, \cdots, \mathcal{M}_{Ie}^{(n_a)}\}$.

The defined inter-attribute coupling mapping function reveals the value frequency in each subspace spanned by values in other attributes. The distance defined in the inter-attribute coupling space reflects inter-attribute couplings. The dimensionality of the inter-attribute coupling space equals $|V| - |V^{(j)}|$. The degree of freedom of the j-th attribute is $|V^{(j)}| - 1$, which implies that the inter-attribute mapping function can project categorical values into a higher dimensional space if $|V| > 2|V^{(j)}| - 1$. In this case, the inter-attribute coupling space is powerful enough to describe the categorical attribute space.

Learning Attribute-Class Couplings. The *attribute-class couplings* capture the interactions between attributes and classes, which reveal the relationships between attribute value distributions w.r.t. each class. For a categorical value $\mathsf{v}_i^{(j)}$ in the j-th attribute, the learning function $m_{Ac}^{(j)}(\mathsf{v}_i^{(j)})$ adopts ICPF to reveal value distributions w.r.t. classes:

$$m_{Ac}^{(j)}(\mathsf{v}_i^{(j)}) = \begin{bmatrix} p(\mathsf{v}_i^{(j)}|c_1) & \cdots & p(\mathsf{v}_i^{(j)}|c_{n_c}) \end{bmatrix}^\top. \tag{10}$$

The inter-attribute coupling space is a N_c-dimensional space that is spanned by the vector obtained by Eq. (10) as follows:

$$\mathcal{M}_{Ac}^{(j)} = \{m_{Ac}^{(j)}(\mathsf{v})|\mathsf{v} \in V^{(j)}\}. \tag{11}$$

For categorical data with n_a attributes, the attribute-class coupling spaces are $\mathcal{M}_{Ac} = \{\mathcal{M}_{Ac}^{(1)}, \cdots, \mathcal{M}_{Ac}^{(n_a)}\}$.

Let us explain the need to consider attribute-class couplings. Assume the number of students who pass an exam is equal to those who fail the exam, then the similarity between passes and failures is very high when only the intra-attribute coupling is considered, indicating that the two outcomes, pass and fail are similar. In reality, this may not make sense; with the attribute-class couplings, we can obtain a more reasonable understanding. If students are classified as first or second class based on their overall performance, many more passes than fails appear in the first group, compared to fewer passes in the second group. This shows that passes and fails are not highly coupled across all classes and it is necessary to explore attribute-class couplings which can complement intra-attribute couplings.

3.5 Heterogeneity Learning

The learned couplings preserve the basic characteristics and different relationships of categorical values in its generated spaces. Although these couplings reveal low-level value relationships from various aspects, the high-level complex relationships among coupling spaces need an in-depth analysis. Specifically, the heterogeneity in these coupling spaces should be learned, and the interactions among these coupling spaces should be disentangled.

The heterogeneity in coupling spaces refers to the different value distributions and different value relationships. Intrinsically, such heterogeneities are caused by two factors: (1) each attribute may have a different value distribution, and values in an attribute may have different distributions; (2) each coupling may reflect a different kind of relationship. Learning the heterogeneous distributions can capture the difference between attributes and reveal the local structure in each attribute. Meanwhile, learning the heterogeneous relationships can find the couplings that are consistent with the final task while filtering the noise caused by the couplings that deviated from the final task.

The disentangled interactions between coupling spaces assist in discovering the independent components, and combining them can obtain the complete information in coupling spaces without redundancy. Value-to-class couplings capture hierarchical value relationships with regard to other values, attributes, and classes. In order to combine these relationships that are at different levels, a further transformation is needed to map them into the same space. Meanwhile, these couplings are not extremely independent. They may contain consensus and complementary information. To reduce the duplicated information, the coupling spaces should be selectively combined according to the interactions among them.

HELIC learns heterogeneity by adopting a variety of kernels to map heterogeneous coupling spaces into homogeneous kernel spaces and learning an adaptive kernel combination to integrate information with a sparse regularization. The intuition behind this process is that different kernels are sensitive to different distributions. If the impact of a kernel on the values that match its sensitive local distribution can be preserved while the impact on others can be released, heterogeneous distribution learning can be wrapped into kernel learning. Meanwhile, the side information for the final learning task can be used to guide the kernel learning. Therefore, the relationships related to the final task can be learned as well. As an effect of sparse regularization, the duplicated information in a kernel space can be reduced.

Given a kernel function $k(\cdot,\cdot)$ and a coupling space for the j-th attribute $\mathcal{M}^{(j)}$, denoting the vector in the coupling space corresponding to value $\mathsf{v}_i^{(j)}$ as m_i, i.e., $\mathbf{m}_i = m^{(j)}(\mathsf{v}_i^{(j)})$, the kernel space is constructed by mapping each value pair as follows,

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{m}_1, \mathbf{m}_1) & k(\mathbf{m}_1, \mathbf{m}_2) & \cdots & k(\mathbf{m}_1, \mathbf{m}_{n_v^{(j)}}) \\ k(\mathbf{m}_2, \mathbf{m}_1) & k(\mathbf{m}_2, \mathbf{m}_2) & \cdots & k(\mathbf{m}_2, \mathbf{m}_{n_v^{(j)}}) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{m}_{n_v^{(j)}}, \mathbf{m}_1) & k(\mathbf{m}_{n_v^{(j)}}, \mathbf{m}_2) & \cdots & k(\mathbf{m}_{n_v^{(j)}}, \mathbf{m}_{n_v^{(j)}}) \end{bmatrix}.$$

Using various kernel functions for the value-to-class coupling spaces, a set of kernel matrices $\{\mathbf{K}_1, \cdots, \mathbf{K}_{n_k}\}$ can be obtained.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2791525, IEEE Transactions on Knowledge and Data Engineering

Further, a set of transformation matrices $\{\mathbf{T}_1, \cdots, \mathbf{T}_{n_k}\}$ can be learned to guarantee that the space of the p-th transformed kernel $\mathbf{K}'_p$ only contains the p-th kernel sensitive information, where $\mathbf{K}'_p$ is defined as:

$$\mathbf{K}'_p = \mathbf{T}_p \cdot \mathbf{K}_p. \tag{12}$$

HELIC forces the transformation matrix $\mathbf{T}_p$ to be a diagonal matrix. In this case, the diagonal values of $\mathbf{T}_p$ are the weights of each categorical value for the p-th kernel. Let $\mathbf{T}_{p,ij}$ denote the value of the (i, j)-th entry of matrix $\mathbf{T}_p$. The large $\mathbf{T}_{p,ii}$ implies that the p-th kernel is more sensitive to the i-th value. Consequently, the spaces spanned by these transformed kernels are heterogeneous kernel spaces. To capture the relationships within value-to-class coupling spaces and enhance the fitness of the distance measure in heterogeneous kernel spaces, HELIC wraps the above heterogeneity learning into metric learning to comprehensively learn the kernel transformation matrix and the distance measure. These will be discussed in the next section.

3.6 Metric Learning

To learn a suitable distance measure for data in heterogeneous kernel spaces, HELIC uses the squared Euclidean distance in each heterogeneous kernel space as the base distance measure, and then combines them to construct a suitable distance measure in heterogeneous kernel spaces.

Given a categorical data set, considering the p-th kernel matrix corresponding to the q-th attribute, let i and j represent the index of values in the p-th kernel space corresponding to the i-th and j-th objects respectively. Specifically, $\mathsf{v}_\mathrm{i}^{(q)} = v^{(q)}(\mathsf{o}_i)$ and $\mathsf{v}_\mathrm{j}^{(q)} = v^{(q)}(\mathsf{o}_j)$. The distance between o_i and o_j in the p-th heterogeneous kernel space is $d_{p,ij}$:

$$d_{p,ij} = (\mathbf{K}'_{p,\mathrm{i}\cdot} - \mathbf{K}'_{p,\mathrm{j}\cdot})^\top (\mathbf{K}'_{p,\mathrm{i}\cdot} - \mathbf{K}'_{p,\mathrm{j}\cdot}), \tag{13}$$

where $\mathbf{K}'_{p,\mathrm{i}\cdot}$ and $\mathbf{K}'_{p,\mathrm{j}\cdot}$ are the i-th and j-th columns of $\mathbf{K}'_p$, respectively. As shown in Eq. (13), the base distance $d_{p,ij}$ is determined by both the given kernel and the learned transformation matrix. In addition, it equals a squared Mahalanobis distance in its original kernel space,

$$d_{p,ij} = (\mathbf{K}_{p,\mathrm{i}\cdot} - \mathbf{K}_{p,\mathrm{j}\cdot})^\top \mathbf{T}_p^\top \mathbf{T}_p (\mathbf{K}_{p,\mathrm{i}\cdot} - \mathbf{K}_{p,\mathrm{j}\cdot}). \tag{14}$$

The distance metric d_{ij} between the i-th and j-th objects is defined by a linear combination of base distance measures from heterogeneous kernel spaces:

$$d_{ij} = \sum_{p=1}^{n_k} \alpha_p d_{p,ij}, \tag{15}$$

where α_p is the weight for the p-th base distance measure.

With a positive semi-definite matrix $\boldsymbol{\omega}_p = \alpha_p \mathbf{T}_p^\top \mathbf{T}_p$, the metric d_{ij} is calculated as:

$$d_{ij} = \sum_{p=1}^{n_k} \mathbf{k}_{p,ij}^\top \boldsymbol{\omega}_p \mathbf{k}_{p,ij}, \tag{16}$$

where $\mathbf{k}_{p,ij} = \mathbf{K}_{p,\mathrm{i}\cdot} - \mathbf{K}_{p,\mathrm{j}\cdot}$. Further, we define a vector,

$$\mathbf{k}_{ij} = \begin{bmatrix} \mathbf{k}_{1,ij}^\top & \mathbf{k}_{2,ij}^\top & \cdots & \mathbf{k}_{n_k,ij}^\top \end{bmatrix}^\top, \tag{17}$$

and a diagonal matrix,

$$\boldsymbol{\omega} = \begin{bmatrix} \boldsymbol{\omega}_1^{\mathrm{diag}} & 0 & \cdots & 0 \\ 0 & \boldsymbol{\omega}_2^{\mathrm{diag}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{\omega}_{n_k}^{\mathrm{diag}} \end{bmatrix}, \tag{18}$$

where $\boldsymbol{\omega}_p^{\mathrm{diag}}$ is the diagonal matrix of $\boldsymbol{\omega}_p$. The metric d_{ij} is equal to a Mahalanobis distance in $\mathbf{k}_{ij}$'s space with a positive semi-definite matrix $\boldsymbol{\omega}$.

$$d_{ij} = \mathbf{k}_{ij}^\top \boldsymbol{\omega} \mathbf{k}_{ij}. \tag{19}$$

The learning of the set of kernel transformation matrices $\{\mathbf{T}_1, \cdots, \mathbf{T}_{n_k}\}$ and the combination coefficient of base distance measures $\{\alpha_1, \cdots, \alpha_{n_k}\}$ can be wrapped into the learning of a positive semi-definite matrix $\boldsymbol{\omega}$. In other words, to construct the metric space for categorical data, we only need to learn the positive semi-definite matrix $\boldsymbol{\omega}$. Consequently, the data vector of the i-th object in the learned metric space can be represented by

$$\mathbf{x}_i = [\sqrt{\boldsymbol{\omega}_{1,11}}\mathbf{K}_{1,\mathrm{i}1}, \cdots, \sqrt{\boldsymbol{\omega}_{n_k,n_v^* n_v^*}}\mathbf{K}_{n_k,\mathrm{i}n_v^*}], \tag{20}$$

where $\boldsymbol{\omega}_{i,jj}$ means the value of the (j, j)-th entry in $\boldsymbol{\omega}_i$, and n_v^* refers to the number of values in the attribute corresponding to the n_k-th kernel. The learned representation $\mathbf{x}_i$ can be fed into a vector-based classifier as a numerical approximation of categorical data. Due to space limitations, this paper only uses labels as the side information and assumes that the distance between objects with the same label is smaller than the distance between objects with different labels. $\boldsymbol{\omega}$ should be sparse because each kernel is only sensitive to partial structure.

The learning objective function is defined as:

$$\begin{aligned} \underset{\boldsymbol{\omega}, b}{\text{minimize}} \quad & \frac{1}{n_o^2} \sum_{i,j \in N_o} \xi_{ij} + \lambda \|\boldsymbol{\omega}\|_1 \\ \text{subject to} \quad & \boldsymbol{\omega} \succcurlyeq 0, \\ & \boldsymbol{\omega}_{kl} = 0 \quad for \quad k \neq l, \\ & 1 + r_{ij}(\mathbf{k}_{ij}^\top \boldsymbol{\omega} \mathbf{k}_{ij} - b) \leqslant \xi_{ij}, \\ & \xi_{ij} \geqslant 0, \forall i, j \in N_o. \end{aligned} \tag{21}$$

In the above function, $\|\cdot\|_1$ refers to the ℓ_1-norm, λ is a trade-off parameter that balances empirical error and regularization, and $\boldsymbol{\omega}_{kl}$ refers to the value of the (k, l)-th entry in the matrix $\boldsymbol{\omega}$. r_{ij} indicates whether two objects have the same label, defined as:

$$r_{ij} = \begin{cases} 1, & c(\mathsf{o}_i) = c(\mathsf{o}_j) \\ -1, & c(\mathsf{o}_i) \neq c(\mathsf{o}_j) \end{cases} \tag{22}$$

Since $\boldsymbol{\omega}$ is a diagonal matrix, this objective function can be efficiently optimized by linear programming.

In this paper, we use the stochastic optimization method to obtain an approximate optimal solution for the learning objective function. As shown in Eq. (21), the number of training data is n_o^2 if there are n_o categorical objects. Although linear programming can be used to find the global optimal of Eq. (21), it is time and space consuming when the amount of data is large. Instead of training all the data at once, HELIC uses the stochastic optimization method to randomly select a mini-batch of object pairs to calculate gradient values and update the learning parameters at each iteration. After several iterations of training, HELIC reaches an approximate optimal that can construct the metric for categorical data. Section 4.3 will theoretically analyze the necessity and effectiveness of using stochastic optimization to obtain the approximate solution, and Section 5.6 will empirically evaluate the convergence of stochastic optimization for solving the objective function. Both theoretical and empirical analyses demonstrate that the stochastic optimization method can effectively solve the objective function with a fast convergence speed. Therefore, HELIC enjoys good scalability for large-scale categorical metric learning.

4 Theoretical Analysis of HELIC Performance

4.1 HELIC Effectiveness

Before proving the effectiveness of HELIC for categorical data, we give the following lemma.

Lemma 1. Given an ideal kernel space that is spanned by a set of kernels $\mathcal{K} = \{\mathbf{K}_1, \cdots, \mathbf{K}_n\}$ to capture various information, the target similarity in categorical space $\mathcal{S}$, and a bounded loss function $\ell(\cdot)$, the expected loss of similarity with respect to $\mathcal{S}$ based on $\mathcal{K}$ and its subset $\mathcal{K}_*$ are denoted by $\hat{\ell}(\mathcal{S};\mathcal{K})$ and $\hat{\ell}(\mathcal{S};\mathcal{K}_*)$. Their difference is bounded as

$$|\hat{\ell}(\mathcal{S};\mathcal{K}) - \hat{\ell}(\mathcal{S};\mathcal{K}_*)| \le \sqrt{I(\mathcal{S};\mathcal{K}\setminus\mathcal{K}_*|\mathcal{K}_*)}, \tag{23}$$

where $I(\cdot;\cdot|\cdot)$ is the conditional mutual information, and $\mathcal{K}\setminus\mathcal{K}_*$ refers to the subspace that is spanned by the set of kernels in $\mathcal{K}$ minus the set of kernels in $\mathcal{K}_*$.

Proof. Considering $|\ell(x)| \le 1$, and two probability distributions P and Q, we have

$$\begin{aligned}\left|\int \ell(x)dQ - \int \ell(x)dP\right| &= \left|\int (1-\theta)\ell(x)dQ\right| \\ &\le \int |1-\theta|\, dQ \\ &\le \sqrt{KL(Q;P)},\end{aligned} \tag{24}$$

where $\theta = \frac{dP}{dQ}$, $KL(\cdot;\cdot)$ is the KL divergence, and the upper bound holds because the ℓ_1 variational distance is bounded by the square root of the KL divergence.

With the above, for a fixed ideal similarity set $\mathcal{K}$, its subspace $\mathcal{K}_*$ and the target similarity space $\mathcal{S}$, we have

$$|E_{\mathcal{S}|\mathcal{K}}\ell(\mathcal{S};\mathcal{K})| - E_{\mathcal{S}|\mathcal{K}_*}\ell(\mathcal{S};\mathcal{K}) \le \sqrt{KL(P_{\mathcal{S}|\mathcal{K}};P_{\mathcal{S}|\mathcal{K}_*})}, \tag{25}$$

where $P_{\mathcal{S}|\mathcal{K}}$ and $P_{\mathcal{S}|\mathcal{K}_*}$ are the conditional distribution of $\mathcal{S}$ conditioned on $\mathcal{K}$ and $\mathcal{K}_*$, respectively.

Taking the expectation with respect to $\mathcal{K}$ and Jensen's inequality, we obtain

$$|\hat{\ell}(\mathcal{S};\mathcal{K}) - E_{\mathcal{K}}E_{\mathcal{S}|\mathcal{K}_*}\ell(\mathcal{S};\mathcal{K})| \le \sqrt{E_{\mathcal{K}}KL(P_{\mathcal{S}|\mathcal{K}};P_{\mathcal{S}|\mathcal{K}_*})}. \tag{26}$$

Since

$$\hat{\ell}(\mathcal{S};\mathcal{K}_*) \le E_{\mathcal{K}}E_{\mathcal{S}|\mathcal{K}_*}\ell(\mathcal{S};\mathcal{K}) \tag{27}$$

and

$$\hat{\ell}(\mathcal{S};\mathcal{K}) \le \hat{\ell}(\mathcal{S};\mathcal{K}_*), \tag{28}$$

we obtain

$$|\hat{\ell}(\mathcal{S};\mathcal{K}) - \hat{\ell}(\mathcal{S};\mathcal{K}_*)| \le \sqrt{E_{\mathcal{K}}KL(P_{\mathcal{S}|\mathcal{K}}, P_{\mathcal{S}|\mathcal{K}_*})}. \tag{29}$$

Considering

$$E_{\mathcal{K}}KL(P_{\mathcal{S}|\mathcal{K}}, P_{\mathcal{S}|\mathcal{K}_*}) = I(\mathcal{S};\mathcal{K}\setminus\mathcal{K}_*|\mathcal{K}_*), \tag{30}$$

we have

$$|\hat{\ell}(\mathcal{S};\mathcal{K}) - \hat{\ell}(\mathcal{S};\mathcal{K}_*)| \le \sqrt{I(\mathcal{S};\mathcal{K}\setminus\mathcal{K}_*|\mathcal{K}_*)}. \tag{31}$$

□

Therefore, HELIC effectiveness is ensured by the following theorem.

Theorem 4.1. *HELIC can improve the metric accuracy if each coupling space contains information complementary to the other coupling spaces and the heterogeneous kernel spaces capture such information effectively.*

Proof. Given a set of coupling space

$$\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_{n^*}\},$$

and a set of kernel matrices generated on the coupling space $\{\mathbf{K}_1, \cdots, \mathbf{K}_n\}$, let the kernel space spanned by these kernel matrices be $\mathcal{K}_* = \{\mathbf{K}_1, \cdots, \mathbf{K}_n\}$ and the target similarity space $\mathcal{S}$. The kernel space $\mathcal{K}_*$ must belong to an ideal space $\mathcal{K}$ which obeys the following property.

$$I(\mathcal{S};\mathcal{K}_m|\mathcal{K}_{m-1}) > 0, \tag{32}$$

where $\mathcal{K}_m$ and $\mathcal{K}_{m-1}$ refer to the subspace in $\mathcal{K}$ that are spanned by a set of m components and a subset with $m-1$ elements in that set, respectively.

In addition,

$$\begin{aligned}&I(\mathcal{S};\mathcal{K}\setminus\mathcal{K}_{m-1}|\mathcal{K}_{m-1}) - I(\mathcal{S};\mathcal{K}\setminus\mathcal{K}_m|\mathcal{K}_m)\\ &= (H(\mathcal{S},\mathcal{K}_{m-1}) + H(\mathcal{K}) - H(\mathcal{S},\mathcal{K}) - H(\mathcal{S},\mathcal{K}_{m-1}))\\ &- (H(\mathcal{S},\mathcal{K}_m) + H(\mathcal{K}) - H(\mathcal{S},\mathcal{K}) - H(\mathcal{S},\mathcal{K}_m))\\ &= (H(\mathcal{K}_m) - H(\mathcal{K}_{m-1})) - (H(\mathcal{S},\mathcal{K}_m) - H(\mathcal{S},\mathcal{K}_{m-1}))\\ &= H(\mathcal{K}_m|\mathcal{K}_{m-1}) - H(\mathcal{K}_m|\mathcal{S},\mathcal{K}_{m-1})\\ &= I(\mathcal{S},\mathcal{K}_m|\mathcal{K}_{m-1}).\end{aligned} \tag{33}$$

Therefore, we have

$$I(\mathcal{S};\mathcal{K}\setminus\mathcal{K}_{m-1}|\mathcal{K}_{m-1}) - I(\mathcal{S};\mathcal{K}\setminus\mathcal{K}_m|\mathcal{K}_)) > 0. \tag{34}$$

According to Lemma 1 and Eq. (34), when m increases, $I(\mathcal{S};\mathcal{K}\setminus\mathcal{K}_m|\mathcal{K}_m)$ as well as the gap between $\ell(\mathcal{S};\mathcal{K})$ and $\ell(\mathcal{S};\mathcal{K}_m)$ will decrease. Therefore, the similarities in the kernel space $\mathcal{K}_*$ can increase the similarity accuracy. This means that HELIC can improve the similarity accuracy if each coupling space contains information that is complementary to others. □

Since HELIC adopts different coupling learning functions to capture data characteristics w.r.t. the intra-attribute value distributions, inter-attribute interactions, and attribute-class relationships, the information contained in each coupling space reflects a respective view and is thus complementary to other coupling spaces. The proposed heterogeneity learning approach reveals and combines different information from value-to-class coupling spaces. Hence, HELIC-based metric learning captures heterogeneous characteristics embedded in categorical data.

4.2 HELIC's Generalization Error Bound

This section analyzes HELIC's learning generalization error bound which shows to what extent HELIC can achieve its effectiveness. The HELIC's generalization error bound is given by the following theorem.

Theorem 4.2. *Let $\varepsilon(\boldsymbol{\omega}, b)$ be the expected error, and $\varepsilon_{\mathcal{Z}}(\boldsymbol{\omega}, b)$ be the empirical error. For any $0 < \delta < 1$, the generalization error of learning can be bounded with probability $1-\delta$ as*

$$\begin{aligned}\varepsilon(\boldsymbol{\omega}, b) - \varepsilon_{\mathcal{Z}}(\boldsymbol{\omega}, b) &\leqslant 2(1 + 1/\sqrt{\lambda})\sqrt{2\ln(1/\delta)/n_o}\\ &+ \left(8 + 16\sqrt{e\ln(n_o n_k)}\right)/\sqrt{n_o\lambda} + 12/\sqrt{n_o}.\end{aligned} \tag{35}$$

Proof. According to Theorem 3 in [36], we have

$$\begin{aligned}\varepsilon\left(\boldsymbol{\omega},b\right)-\varepsilon_{\mathcal{Z}}\left(\boldsymbol{\omega},b\right)\leqslant&\frac{4R_n}{\sqrt{\lambda}}+\frac{4(3+2X_*/\sqrt{\lambda})}{\sqrt{n_o}}\\&+2\left(1+X_*/\sqrt{\lambda}\right)\left(\frac{2\ln\left(\frac{1}{\sigma}\right)}{n_o}\right)^{\frac{1}{2}},\end{aligned}\tag{36}$$

where R_n is the Rademacher complexity of metric learning defined as

$$R_n=\frac{1}{\left\lfloor\frac{n_o}{2}\right\rfloor}E_{\mathcal{K}}E_{\sigma}\left\|\sum_{i=1}^{\left\lfloor\frac{n_o}{2}\right\rfloor}\sigma_i\mathbf{k}_{i\left(\left\lfloor\frac{n_o}{2}\right\rfloor+1\right)}\right\|_*,$$

and

$$X_*=\sup_{\mathbf{k}\in\mathcal{K}}\left\|\mathbf{k}_{ij}\mathbf{k}_{ij}^{\top}\right\|_*.$$

In the above formula, $\|\cdot\|_*$ refers to the dual norm of ℓ_1-norm, $\left\lfloor\frac{n_o}{2}\right\rfloor$ denotes the largest integer less than $\frac{n_o}{2}$, $\mathcal{K}$ is the kernel space, $\{\sigma_1,\sigma_2,\cdots,\sigma_{\lfloor\frac{n_o}{2}\rfloor}\}$ are independent variables drawn from the Rademacher distribution, and $E_{\mathcal{D}}$ is the expectation over space $\mathcal{D}$. Since the dual norm of ℓ_1-norm is the ℓ_∞-norm and the maximum of $\mathbf{k}_{ij}$ in the space $\mathcal{K}$ is not more than 1, hence

$$X_*=\sup_{\mathbf{k}\in\mathcal{K}}\|\mathbf{k}_{ij}\|_\infty^2\leqslant 1,\tag{37}$$

and the Rademacher complexity can be written as

$$R_n=\frac{1}{\left\lfloor\frac{n_o}{2}\right\rfloor}E_{\mathcal{K}}E_{\sigma}\left\|\sum_{i=1}^{\left\lfloor\frac{n_o}{2}\right\rfloor}\sigma_i\mathbf{k}_{i\left(\left\lfloor\frac{n_o}{2}\right\rfloor+i\right)}\right\|_\infty.$$

Considering the property of norm and letting $i^*=\left(\left\lfloor\frac{n_o}{2}\right\rfloor+i\right)$, for any $1<q<\infty$, that

$$\begin{aligned}R_n&\leqslant\frac{1}{\left\lfloor\frac{n_o}{2}\right\rfloor}E_{\mathcal{K}}E_{\sigma}\left\|\sum_{i=1}^{\left\lfloor\frac{n_o}{2}\right\rfloor}\sigma_i\mathbf{k}_{ii^*}\right\|_q\\&=\frac{1}{\left\lfloor\frac{n_o}{2}\right\rfloor}E_{\mathcal{K}}E_{\sigma}\left(\sum_{l=1}^{n_on_k}\sum_{m=1}^{n_on_k}\left|\sum_{i=1}^{\left\lfloor\frac{N_o}{2}\right\rfloor}\sigma_i\mathbf{k}_{l,ii^*}\mathbf{k}_{m,ii^*}\right|^q\right)^{\frac{1}{q}}\\&\leqslant\frac{1}{\left\lfloor\frac{n_o}{2}\right\rfloor}E_{\mathcal{K}}\left(\sum_{l=1}^{n_on_k}\sum_{m=1}^{n_on_k}E_{\sigma}\left|\sum_{i=1}^{\left\lfloor\frac{n_o}{2}\right\rfloor}\sigma_i\mathbf{k}_{l,ii^*}\mathbf{k}_{m,ii^*}\right|^q\right)^{\frac{1}{q}},\end{aligned}\tag{38}$$

where $\mathbf{k}_{l,ij}$ refers to the l-th element of vector $\mathbf{k}_{ij}$ defined as in Eq. (17). Considering the Khinchin-Kahane inequality [37], that

$$\begin{aligned}&E_{\sigma}\left|\sum_{i=1}^{\left\lfloor\frac{n_o}{2}\right\rfloor}\sigma_i\mathbf{k}_{l,ii^*}\mathbf{k}_{m,ii^*}\right|^q\\&\leqslant q^{\frac{q}{2}}\left(E_{\sigma}\left|\sum_{i=1}^{\left\lfloor\frac{n_o}{2}\right\rfloor}\sigma_i\mathbf{k}_{l,ii^*}\mathbf{k}_{m,ii^*}\right|^2\right)^{\frac{q}{2}}\\&=q^{\frac{q}{2}}\left(\sum_{i=1}^{\left\lfloor\frac{n_o}{2}\right\rfloor}\mathbf{K}_{l,ii^*}^2\mathbf{K}_{m,ii^*}^2\right)^{\frac{q}{2}}\\&\leqslant\sup_{\mathbf{k}\in\mathcal{K}}\|\mathbf{k}_{ij}\|_\infty^{2q}\left(\left\lfloor\frac{n_o}{2}\right\rfloor\right)^{\frac{q}{2}}q^{\frac{q}{2}}\\&\leqslant\left(\left\lfloor\frac{n_o}{2}\right\rfloor\right)^{\frac{q}{2}}q^{\frac{q}{2}}.\end{aligned}\tag{39}$$

Putting the result of Eq. (39) into Eq. (38), we obtain that

$$R_n\leqslant\frac{1}{\left\lfloor\frac{n_o}{2}\right\rfloor}\left((n_on_k)^2\left(\left\lfloor\frac{n_o}{2}\right\rfloor\right)^{\frac{q}{2}}q^{\frac{q}{2}}\right)^{\frac{1}{q}}.\tag{40}$$

Letting $q=4\ln(n_on_k)$, Eq. (40) induces that

$$\begin{aligned}R_n&\leqslant 2\sqrt{e\ln(n_on_k)/\left\lfloor\frac{n_o}{2}\right\rfloor}\\&\leqslant 4\sqrt{e\ln(n_on_k)/n_o}.\end{aligned}\tag{41}$$

Putting Eq. (37) and Eq. (41) into Eq. (36) yields Eq. (35). Theorem 4.2 is thus proved. □

This bound provides a solid foundation for HELIC and offers the following guidance. (1) More training data leads to a lower generalization error. (2) The smaller the number of base kernels, the lower the generalization error. Fortunately, d has a logarithmic relationship with the generalization error. This shows that the number of base kernels will not dramatically increase the generalization error. (3) According to Theorem 4.1, increasing the number of base kernels provides more complementary information which reduces the expected error $\varepsilon(\boldsymbol{\omega},b)$. Therefore, the overall performance will still increase when complementary base kernels are added. This can be achieved by building additional coupling spaces and using more discriminative kernels.

4.3 Computational Efficiency

The HELIC time complexity is determined by two parts: coupling learning and metric learning. For the coupling learning part, the time cost depends on what kind of couplings HELIC captures. In this paper, HELIC captures the intra-attribute couplings, inter-attribute couplings, and attribute-class couplings. Calculating intra-attribute coupling requires the measurement of the frequency of each value, which has a complexity $O(n_v)$. Capturing inter-attribute couplings requires calculating the relationship of each value pair in each attribute pair. Accordingly, the time cost is $O(n_{mv}^2n_a^2)$, where n_{mv} is the maximal number of values in the attributes. For the attribute-class couplings, HELIC needs to calculate the relation between each value and each class that has time complexity of $O(n_vn_c)$. Therefore, the time complexity for the coupling learning part is $O(n_v(n_c+1)+n_{mv}^2n_a^2)$ in this paper.

In metric learning, the time complexity depends on the solution. If using linear programming to find the global optima, the time complexity in the worst case scenario is $O((n_o^2)^{3.5}n_{\boldsymbol{\omega}}^2)$, and a fast approximate solution can achieve $O(n_o^2 + \frac{2(n_{\boldsymbol{\omega}}+n_o^2)\log(n_o^2)}{\epsilon^2})$ with $O(1 \pm \epsilon)$ cost, where $n_{\boldsymbol{\omega}} = \sum_{i=1}^{n_k} n_v^{(i*)}$ is the length of parameter $\boldsymbol{\omega}$, and $n_v^{(i^*)}$ refers to the number of values in attribute i^* corresponding to the i-th kernel. If using stochastic optimization for an approximate optimum, the time complexity is only $O(n_b n_{\boldsymbol{\omega}} n_{step})$, where n_b is the number of object pairs in each batch and n_{step} is the number of iterations used to achieve convergence. Compared with the linear programming method, stochastic optimization is much more efficient if it can converge within a small number of iterations.

Actually, stochastic optimization can effectively find an approximate optimal solution for HELIC. Considering the structure of our objective function, the key parameter which needs to be learned is $\boldsymbol{\omega}$, which is a vector with size $n_{\boldsymbol{\omega}}$. When $n_o^2 \gg n_{\boldsymbol{\omega}}$, the loss of the objective function will converge before using all n_o^2 training data. Since $n_{\boldsymbol{\omega}} = \sum_{i=1}^{n_k} n_v^{(i*)} = n_v n_k^*$, where n_k^* is a constant that relates to the number of used kernels, the above condition can be rewritten as $n_o^2 \gg n_v n_k^*$. Fortunately, the large scale categorical data always fits this condition because: (1) the number of discrete values is much less than the number of objects; (2) the number of base kernels used in HELIC should be small to guarantee a lower generalization error according to Section 4.2.

The space complexity of HELIC with linear programming is $O(n_o^2 n_{\boldsymbol{\omega}})$ and of HELIC with stochastic optimization is $O(n_b n_{\boldsymbol{\omega}})$. For a large categorical data set, the space complexity is very high when linear programming is used to find a perfect solution. However, a stochastic optimization-based approximate solution largely reduces the space complexity since $n_b \ll n_o^2$. Therefore, it is necessary to use stochastic optimization for HELIC to tackle large categorical data.

Overall, HELIC has the time complexity $O(n_v(n_c + 1) + n_{mv}^2 n_a^2 + n_b n_{\boldsymbol{\omega}} n_{step})$ and space complexity $O(n_b n_{\boldsymbol{\omega}})$. This means HELIC suits large data. In addition, HELIC can be further sped up by applying parallel computing to both coupling learning and metric learning for large data.

5 Experiments and Evaluation

We empirically evaluate the proposed HELIC framework w.r.t. the following five criteria:

1) the HELIC representation performance: whether HELIC enables a model to obtain better results;
2) the HELIC representation quality: the goodness of the metric learned by HELIC;
3) the effect of learning couplings and heterogeneity: to what extent the learned couplings and heterogeneity contribute to the metric;
4) the HELIC scalability: whether HELIC is scalable w.r.t. different data factors for a large amount of data; and
5) the HELIC stability: whether the HELIC performance is stable under different settings.

5.1 Parameter Settings

In our experiments, the HELIC default settings are as follows. The kernels used in HELIC are 11 Gaussian kernels with width from 2^{-5} to 2^5 and three polynomial kernels with order from 1 to 3. λ is set as $\frac{1}{n_k}$ for HELIC. The stochastic optimization method used to solve the HELIC objective function is Adam [38] with the initial learning rate 10^{-3}, the batch size 20, and the number of iterations $1,000$. For the parameters in these baseline methods, we take their recommended settings.

5.2 Data Sets and Characteristics

We use 30 data sets[1] in different areas for the evaluation. These comprise medical data: Lymphography (Lym), Hepatitis (Hep), Audiology (Aud), Primarytumor (Prim), Spect (Spc), Breastcancer (Br), Mammographic (Ma); gene data: DNAPromoter (DNAP), DNANominal (DNAN), Splice (Spc); social and census data: Housevotes (Hsv), Adult (Adt), Census (Cens), Hayesroth (Hay); hierarchical decision-making data: Monks3 (Monk), Krvskp (Krv), Tictactoe (Tic), Krkopt (Krk), Connect4 (Cnt); nature data: Soybeansmall (SoyS), Soybeanlarge (SoyL), Zoo, Flare (Flr), Mushroom (Ms); Business data: Crx; psychological experimental data: Balance (Ba); disaster prediction data: Titanic (Titn); and synthetic data with heterogeneous couplings: Mofn3710 (Mof), ThreeOf9 (Tr) and Led24 (Ld).

These 30 data sets have strong diversity in terms of data factors: the number of objects (n_o), the number of attributes (n_a), the number of classes (n_c), the average number of attribute values (n_{av}), and the maximal number of attribute values (n_{mv}). Specifically, the number of objects ranges from 101 to 299,285, and the number of attributes ranges from 3 to 69. The data sets contain both binary and multiple classes with the maximum number of 24 classes. The average and maximal numbers of attribute values range from 2 to 16.09 and from 2 to 53, respectively. They are used to test the HELIC sensitivity on diverse data characteristics.

Monte Carlo cross-validation is taken to partition a data set to training and test sets. Compared with other validation methods, Monte Carlo cross-validation can largely retain the heterogeneous distributions in the training set. Hence, it is more suitable for evaluating HELIC, which intends to learn the heterogeneous distributions. Specifically, we randomly select 90% of objects in each data set for training and use the remainder for testing, and 20 random sampling iterations generate 20 sets of training and test data for the experiments.

5.3 Testing HELIC Representation Performance

The HELIC representation performance is evaluated from the following perspectives: (1) The HELIC learned metric is compared with the baseline categorical distance measure, i.e., Hamming distance (Hamming for short). (2) It is also compared with five state-of-the-art distance measures for categorical data: COS [23], MTDLE [24], Ahmad [20], DILCA [1], and Rough [18], to show whether HELIC outperforms the others in learning metric.

5.3.1 HELIC-enabled Classification Performance

The distance learned by HELIC is incorporated into KNN, which is probably the most popular distance-based classifier and is sensitive to distance measure, to demonstrate the HELIC-enabled classification performance. To avoid the impact of class imbalance, we evaluate the performance by F-score, which is a combination of recall and precision. A higher F-score indicates better learning accuracy. For a method, the averaged rank (AR) over all data sets is used to evaluate its overall performance.

[1] They are downloaded from: http://archive.ics.uci.edu/ml; https://www.sgi.com/tech/mlc/db; and https://www.kaggle.com.

TABLE 3
KNN Classification F-score (%) with Different Distance Measures. The Monte Carlo cross-validation results are reported w.r.t. *mean* ± *standard deviation*. The best results are highlighted in bold, the results without a significant difference from the best results for a data set under the *student t-test* (*p*-value > 0.05) are labelled by *, and Δ is the HELIC's improvement over the best results of the other measures. The averaged rank of a method over all data sets with significant difference from others w.r.t. the *Bonferroni-Dunn test* (*p*-value < 0.05) is labelled by **.

Data	n_o	n_a	HELIC	COS	MTDLE	Ahmad	DILCA	Rough	Hamming	Δ
SoyS	47	35	**100 ± 0.00***	**100 ± 0.00***	**100 ± 0.00***	**100 ± 0.00** *	**100 ± 0.00***	**100 ± 0.00** *	**100 ± 0.00***	0.00%
Zoo	101	16	**100 ± 0.00***	**100 ± 0.00***	**100 ± 0.00***	**100 ± 0.00***	**100 ± 0.00***	97.75 ± 11.11	**100 ± 0.00***	0.00%
DNAP	106	57	**92.90 ± 5.85***	75.89 ± 13.35	81.67 ± 10.19	79.98 ± 9.14	90.33 ± 10.31	81.16 ± 10.30	78.05 ± 12.00	2.85%
Hay	132	4	**90.85 ± 5.07***	79.64 ± 9.71	68.54 ± 10.55	52.26 ± 10.20	54.60 ± 12.58	81.50 ± 8.59	61.73 ± 12.40	11.47%
Lym	148	18	**86.74 ± 8.11***	77.82 ± 10.01	80.54 ± 10.49	83.84 ± 10.57*	84.32 ± 9.59*	81.25 ± 8.21*	78.52 ± 8.70	2.87%
Hep	155	13	**74.70 ± 13.59***	64.05 ± 13.00	69.17 ± 15.65*	65.40 ± 13.25	61.73 ± 14.22	64.01 ± 14.89	65.65 ± 15.19	7.84%
Aud	200	69	**75.44 ± 7.60***	41.51 ± 7.20	36.70 ± 7.50	54.29 ± 8.96	64.83 ± 8.04	36.37 ± 7.60	58.55 ± 10.30	16.36%
Hsv	232	16	**96.65 ± 3.40**	94.28 ± 4.95	91.09 ± 5.55	95.81 ± 4.15	94.90 ± 4.14	91.59 ± 5.14	93.77 ± 5.30	0.88%
Spc	267	22	**53.09 ±10.35***	51.31 ± 9.16*	52.94 ± 9.48*	52.70 ± 9.69*	51.11 ± 8.97*	51.18 ± 7.90*	51.98 ± 8.85*	0.28%
Mof	300	10	**94.39 ±5.86***	79.35 ± 9.07	68.74 ± 10.58	79.35 ± 9.07	71.21 ± 8.42	77.70 ± 11.44	74.82 ± 8.08	18.95%
SoyL	307	35	90.97 ± 7.06	**93.45 ± 4.87***	64.92 ± 10.09	93.43 ± 4.95*	92.87 ± 5.35*	90.05 ± 4.92	89.84 ± 7.21	0.00%
Prim	339	17	**35.76 ± 8.61***	23.09 ± 6.71	27.00 ± 6.80	28.30 ± 7.93*	27.46 ± 7.56*	20.80 ± 6.64	29.42 ± 9.53	21.55%
Monk	432	6	**100 ± 0.00***	34.85 ± 0.00	99.88 ± 0.52*	34.85 ± 0.00	34.85 ± 0.00	**100 ± 0.00***	92.06 ± 5.24	0.00%
Tr	512	9	**91.01 ±2.93***	32.00 ± 0.00	75.88 ± 8.41	32.00 ± 0.00	32.00 ± 0.00	78.84 ± 5.09	78.84 ± 5.09	15.44%
Ba	625	4	**58.91 ±1.31***	21.25 ± 0.00	41.80 ± 5.82	21.25 ± 0.00	21.25 ± 0.00	39.32 ± 4.25	39.32 ± 4.25	40.93%
Crx	690	9	**83.26 ± 5.68***	78.58 ± 4.74	77.54 ± 5.68	82.79 ± 3.86*	81.02 ± 4.08	77.63 ± 5.12	78.28 ± 4.87	0.57%
Br	699	9	95.72 ± 2.07*	94.07 ± 2.84	93.44 ± 3.21	**96.34 ± 2.00***	94.14 ± 2.50	92.65 ± 3.42	93.93 ± 2.33	0.00%
Ma	830	4	**79.61 ±4.59***	70.22 ± 7.12*	70.14 ± 7.10*	70.20 ± 7.02*	70.22 ± 7.81*	69.79 ± 7.11 *	69.95 ± 7.29*	13.37%
Tic	958	9	92.80 ± 3.49	90.56 ± 2.70	78.29 ± 5.55	**100 ± 0.00***	89.72 ± 3.79	46.65 ± 4.10	46.56 ± 4.65	0.00%
Flr	1066	11	**59.88 ± 3.36***	57.01 ± 4.38*	57.11 ± 3.09	54.41 ± 3.39	55.61 ± 3.13	55.88 ± 4.38	54.98 ± 4.00	4.85%
Titn	2201	3	**23.33 ± 2.48***	10.54 ± 1.76	10.06 ± 0.62	10.06 ± 0.99	10.54 ± 1.76	10.54 ± 1.76	10.54 ± 1.76	32.48 %
DNAN	3186	60	**93.12 ± 1.05***	77.52 ± 1.21	52.22 ± 0.00	80.33 ± 1.48	91.65 ± 1.39	81.46 ± 1.75	69.11 ± 1.45	1.60 %
Spl	3190	60	**93.69 ± 1.11***	77.25 ± 2.19	24.45 ± 0.00	79.85 ± 2.07	84.96 ± 2.21	81.05 ± 1.81	69.29 ± 2.24	10.28 %
Krv	3196	36	**96.98 ± 1.06***	91.77 ± 1.66	90.04 ± 1.65	92.46 ± 1.74	91.39 ± 2.05	89.00 ± 1.43	91.48 ± 1.68	4.89%
Ld	3200	24	**63.37 ± 1.94***	62.11 ± 1.85*	41.35 ± 2.74	61.81 ± 1.98*	62.58 ± 1.85*	47.89 ± 2.37	41.57 ± 2.19	1.26 %
Ms	5644	22	**100 ± 0.00***	99.98 ± 0.06*	**100 ± 0.00***	**100 ± 0.00** *	**100 ± 0.00***	**100 ± 0.00** *	**100 ± 0.00***	0.00%
Krk	28056	6	**53.62 ± 1.71***	52.66 ± 0.78*	NA	52.50 ± 0.96*	52.57 ± 1.02*	39.05 ± 0.70	10.42 ± 0.10	1.82%
Adt	30162	8	**84.91 ± 0.86***	68.13 ± 1.12	NA	68.20 ± 1.07	68.16 ± 1.14	67.76 ± 1.04	68.01 ± 1.04	24.50%
Cnt	67557	42	**56.33 ± 0.78***	48.23 ± 0.73	NA	46.95 ± 0.49	46.65 ± 0.55	53.22 ± 0.73	45.81 ± 0.72	5.84%
Cens	299285	35	**68.93 ± 0.55***	66.88 ± 0.40	NA	67.47 ± 0.43	66.66 ± 0.42	66.96 ± 0.55	67.16 ± 0.37	2.64%
AR	-	-	**1.45****	4.27	4.87	3.73	4.00	4.72	4.68	2.28

HELIC is compared with six distance measures with the results shown in Table 3. The averaged classification performance and standard deviation on the partitions of a data set are reported w.r.t. F-score. The best results are highlighted in bold, the results without significant difference from the best results under the *student t-test* (p-value > 0.05) are labelled by *, and Δ is the HELIC's improvement over the best results of other measures. The averaged ranks of these methods over all data sets are reported to show their overall performance. It should be noted that MTDLE cannot produce the distance results on some large data sets in our experiments due to out-of-memory error, which is marked by *NA* in the table. In most cases, HELIC performs significantly better than the compared methods. For example, the F-score improves 40.93% in Balance and 32.48% in Titanic compared to the best-performing method MTDLE and COS. In some simple data sets, e.g., Zoo, Monks3 and Mushroom, on which the Hamming method achieves high performance, HELIC achieves 100.

The results also show that some of state-of-the-art measures fail to appropriately measure distances in some data sets. For example, on Monks3 and Balance, COS, Ahmad and DILCA achieve the same low performance. The reason lies in the fact that they only capture frequency-based intra- or inter-attribute couplings but ignore other interactions, e.g., attribute-class couplings; values in these data sets follow a uniform distribution and have the same frequency. In contrast, HELIC captures hierarchical value-to-class couplings that address such data characteristics.

To statistically compare HELIC's performance with the above distance measures, we calculate their averaged ranks per the Friedman test and Bonferroni-Dunn test [39]. The χ^2_F of Friedman test is 45.92 associated with p-value $9.42e^{-9}$. This result indicates that the performance of all the compared methods is not equal. Further, the Bonferroni-Dunn test evaluates the critical difference (CD) between HELIC and other methods, and shows the CD at p-value < 0.05 is 1.64. As shown in Table 3, HELIC achieves an overall averaged rank 1.45. Compared with the best state-of-the-art method Ahmad with an averaged rank of 3.73, HELIC improves by 2.28 which is larger than the CD value and shows statistical significance. We show all the comparison results in Fig. 2, which reveals HELIC is significantly ($p < 0.05$) better than all the compared distance measures.

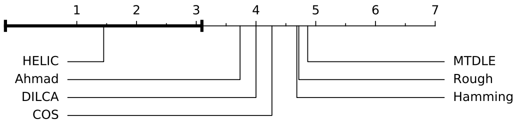
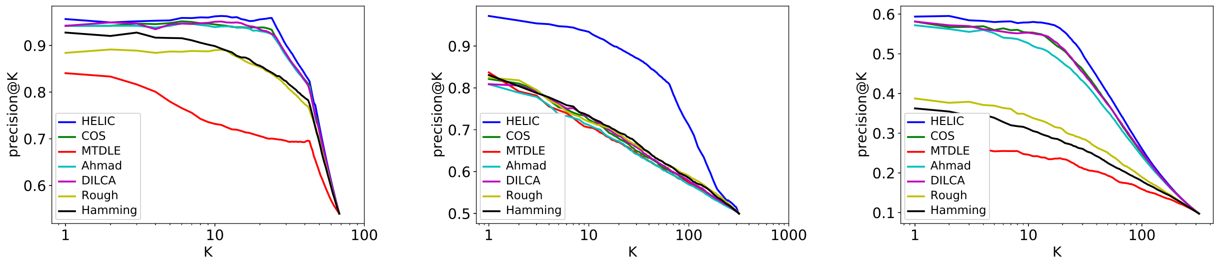


Fig. 2. Comparison of HELIC against the Other Distance Measures per the Bonferroni-Dunn Test. All distance measures with ranks outside the marked interval are significantly different (p < 0.05) from HELIC.

5.3.2 HELIC-enabled Retrieval Performance

We further test the HELIC representation performance in object retrieval. The objects in the test set are used as queries, and precision@k, i.e., the fraction of k-closest objects selected per a distance measure that are the same-class neighbors, is reported. Three data sets, i.e., Breastcancer, Krvskp, and Led24 are tested, in which all the compared distance measures can achieve similar KNN classification accuracy for retrieval performance evaluation.

Different from the KNN classification results, the precision@k of retrieval can demonstrate the quality of learned distance from local (when k is small) to global (when k is large). The results are



(a) Curve on Breastcancer Data Set.

(b) Curve on Krvskp Data Set.

(c) Curve on Led24 Data Set.

Fig. 3. The Precision@k-Curve of Different Distance Measures: A better metric yields a higher curve in this figure.

shown in Fig. 3, the precision of HELIC-enabled retrieval consistently outperforms the others. It reflects that HELIC can capture more details of distribution than other distance measures, which is powered by hierarchical coupling learning and heterogeneity learning.

5.4 Testing HELIC Representation Quality

We follow the definition of (ϵ, γ)-good criterion in [40] to evaluate the representation quality of the proposed HELIC. Following Definition 4 in [40], Q is a strongly (ϵ, γ)-good similarity function for a learning problem P if at least a $1 - \epsilon$ probability mass of objects o satisfies:

$$E_{x'\sim P}[Q(\mathsf{o},\mathsf{o}')|c(\mathsf{o})=c(\mathsf{o}')]$$
$$\geqslant E_{\mathsf{o}'\sim P}[Q(\mathsf{o},\mathsf{o}')|c(\mathsf{o}\neq c(\mathsf{o}'))]+\gamma.$$

The intuition of this criterion is that a good similarity measure should make the data in the same class closer than the data in different classes. More importantly, the (ϵ, γ)-good similarity can induce a classifier with a bounded error (see more details in Theorem 1 of [40]). In this experiment, we transform the learned distance metric D to similarity measure S as follows:

$$S_{ij} = 1 - D_{ij}. \tag{42}$$

The transformed similarity S is then used for the (ϵ, γ)-good criterion. Since different ϵ values may correspond to different γ values, we draw the (ϵ, γ)-curves to demonstrate the quality of the learned metric and compared methods. With the same ϵ, the better metric would have a greater γ. In other words, the better metric would yield a higher curve in the (ϵ, γ)-curve figure. In this experiment, we draw the (ϵ, γ)-curves on three data sets, i.e., DNAPromoter, Audiology and Spect. The results are shown in Fig. 4. It should be noted that we only focus on ϵ that can guarantee a non-negative margin, i.e., $\gamma \geqslant 0$. Therefore, Fig. 4 only displays part of (ϵ, γ)-curve, in which $\gamma \geqslant 0$.

The results illustrate that the proposed HELIC is better than its competitors in terms of (ϵ, γ)-good criterion. The results also reveal the insight behind the KNN classification performance in Table 3. For the DNAPromoter and Audiology data sets, the HELIC-enabled KNN has much higher F-score than the others since HELIC yields a larger margin between different classes, which is reflected by the (ϵ, γ)-good in Figs. 4(a) and 4(b). For the Spect data set, all methods have a low F-score, and the HELIC-enabled KNN is only slightly better than its competitors. This is because none of the methods can well distinguish nearly 40% part of the Spect data, and HELIC can separate more data, compared with other methods, as shown in Fig. 4(c).

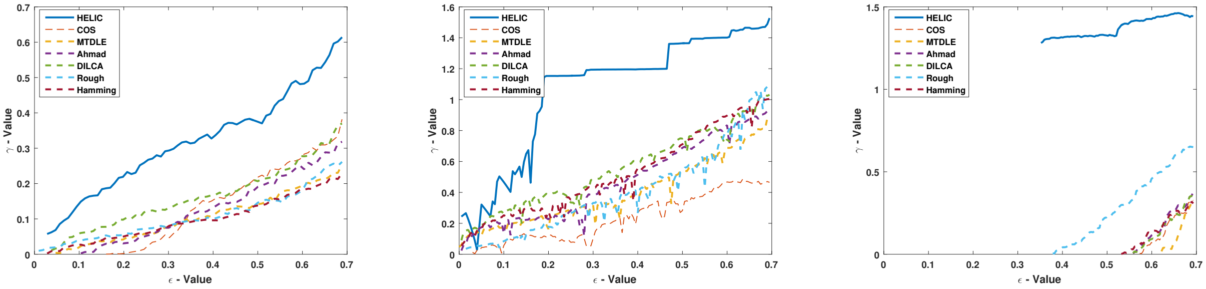
5.5 Effect of Learning Couplings and Heterogeneity

To evaluate the contribution to learning hierarchical couplings and heterogeneity, HELIC is compared with its variant metric HC which only captures hierarchical couplings. HC concatenates on the intra-attribute coupling space, inter-attribute coupling space and attribute-class coupling space for each attribute to construct a value-to-class coupling space. The Euclidean distance in this space is then used as the metric. Meanwhile, HELIC is compared with its variant HELIC-Linear which only adopts a linear kernel to learn a homogeneous metric in the hierarchical coupling space. The comparative classification results are shown in Table 4.

TABLE 4
KNN Classification F-score (%) with HELIC Variants. The Monte Carlo cross-validation results are reported as *mean* ± *standard deviation*. Δ shows the HELIC improvement over the best results of its variants.

Dataset	HELIC	HC	HELIC-Linear	Δ
SoyS	100 ± 0.00	100 ± 0.00	100 ± 0.00	0.00%
Zoo	100 ± 0.00	100 ± 0.00	100 ± 0.00	0%
DNAP	92.90±5.85	94.93 ± 7.00	85.77± 8.75	0%
Hay	90.85±5.07	85.89 ± 6.39	67.09 ± 13.94	5.77%
Lym	86.74 ± 8.11	77.69 ± 12.71	58.98 ± 15.96	11.65%
Hep	74.70 ± 13.59	70.08 ± 13.07	62.27 ± 15.30	6.65 %
Aud	75.44±7.60	54.94 ± 11.85	47.29 ± 7.24	37.31%
Hsv	96.65 ± 3.40	95.43 ± 4.46	94.48 ± 3.90	1.28%
Spc	53.09 ± 10.35	51.40 ± 9.51	50.15 ± 8.18	3.28%
Mfn	94.39 ± 5.86	94.92 ± 3.36	75.16 ± 10.24	0.00%
SoyL	90.97 ± 7.06	92.27 ± 3.86	89.72 ± 5.92	0.00%
Prim	35.76 ± 8.61	26.03 ± 5.82	24.71 ± 5.64	37.38%
Monk	100 ± 0.00	100 ± 0.00	100 ± 0.00	0.00%
Tr	91.01 ± 2.93	89.96 ± 2.92	76.23 ± 5.18	1.17%
Ba	58.91 ± 1.31	59.64 ± 1.46	44.99 ± 7.12	0%
Crx	83.26 ± 5.68	82.43 ± 4.39	81.34 ± 5.17	1.01%
Br	95.72 ± 2.07	94.19 ± 2.80	92.71 ± 2.67	1.62%
Ma	79.61 ± 4.59	70.31 ± 7.00	76.07 ± 8.45	4.65%
Tic	92.80 ± 3.49	79.73 ± 2.60	66.30 ± 4.65	16.39%
Flr	59.88 ± 3.36	55.40 ± 3.93	55.31 ± 4.32	8.09%
Titn	23.33 ± 2.48	12.15 ± 1.65	12.15 ± 1.65	92.02%
DNAN	93.12 ± 1.05	91.83 ± 1.64	87.43 ± 1.34	1.40%
Spc	93.69 ± 1.11	75.88 ± 2.03	63.45 ± 2.60	23.47%
Krv	96.98 ± 1.06	92.49 ± 0.92	95.27 ± 0.82	1.79%
Ld	63.37 ± 1.94	57.71 ± 2.46	51.56 ± 2.11	9.81%
Ms	100 ± 0.00	100 ± 0.00	100 ± 0.00	0.00%
Krk	53.62 ± 1.71	52.44 ± 1.58	52.03 ± 1.96	2.25%
Adt	84.91 ± 0.86	84.32 ± 0.80	68.16 ± 1.46	0.70%
Cnt	56.33 ± 0.78	43.07± 0.50	43.82 ± 0.67	28.55%
Cens	68.93 ± 0.55	64.23 ± 0.49	64.82 ± 0.52	7.32%
AR	**1.27****	2.02	2.72	0.75

We see from Table 4 that the overall averaged rank of HELIC, HC, and HELIC-Linear are 1.27, 1.98 and 2.75, respectively. The Friedman test shows that χ^2_F of these results is 36.76 associated with p-value $1.04e^{-8}$, which means the performance of these three methods is significantly different. We illustrate the comparison between HELIC and its variants in terms of averaged rank in



(a) (ϵ, γ)-curve on DNAPromoter Data Set.

(b) (ϵ, γ)-curve on Audiology Data Set.

(c) (ϵ, γ)-curve on Spect Data Set.

Fig. 4. The (ϵ, γ)-Curve of Different Transformed Similarity Measures: A better metric would yield a curve with higher y-axis values.

Fig. 5, which shows HELIC is significantly better than HC and HELIC-Linear in terms of CD value 0.60 of the Bonferroni-Dunn test at p-value < 0.05. The results demonstrate that heterogeneity learning contributes to an additional 0.75 to the averaged rank of hierarchical couplings. In some data sets, we can see that the F-score improvement ratio of HELIC in terms of HC is very large, e.g., 92.02% on Titanic, and 37.38% on Audiology. However, HELIC does not achieve better performance than HC over all data sets, showing that not all data sets involve strong heterogeneity.

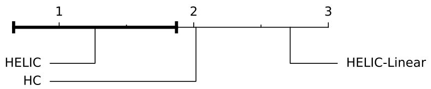


Fig. 5. Comparison of HELIC against Its Variants per the Bonferroni-Dunn Test. All distance measures with ranks outside the marked interval are significantly different (p $<$ 0.05) from HELIC.

The results also demonstrate the significance of learning hierarchical couplings (HC), since HC achieves the averaged rank of 2.88 compared with the other six methods, as shown in Table 3. This is better than that of the best state-of-the-art method Ahmad (3.40). However, HC performance does not show significant difference from others, as shown in Fig. 6.

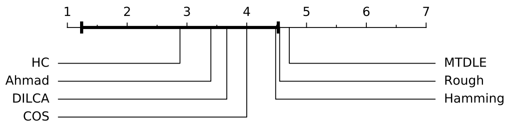


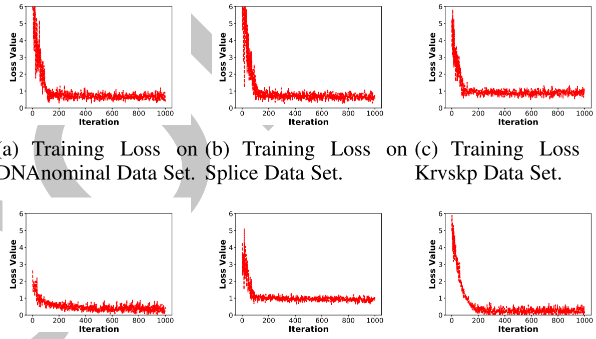
Fig. 6. Comparison of HC against the Other Distance Measures per the Bonferroni-Dunn Test. All distance measures with ranks outside the marked interval are significantly different (p $<$ 0.05) from HC.

The HELIC-Linear performance is worse than that of HC according to Table 4. These results indicate that the distributions in coupling spaces are complex and heterogeneous. Therefore, a learning metric in the linear space transformed by the linear kernel is not sufficient. As analyzed in Section 4.2, a variety of kernels should be tested to capture the heterogeneity.

5.6 Testing HELIC Scalability

HELIC scalability is proportional to the number of iterations to achieve convergence, as discussed in Section 4.3. In this section, we first empirically evaluate the convergence speed of HELIC, and then illustrate the HELIC time cost under different data factors.

We use six real data sets to evaluate HELIC convergence. These data sets represent data with a large number of attributes and objects. The method used to solve HELIC is Adam with the same settings as given in Section 5.1. The loss of objective function Eq. (21) for these data sets is shown in Fig. 7. As shown in Fig. 7, the loss value converges rapidly within 200 iterations. This is consistent with our theoretical analysis and demonstrates that HELIC time complexity is very low.



(a) Training Loss on DNAnominal Data Set.

(b) Training Loss on Splice Data Set.

(c) Training Loss on Krvskp Data Set.

(d) Training Loss on Mushroom Data Set.

(e) Training Loss on Connect4 Data Set.

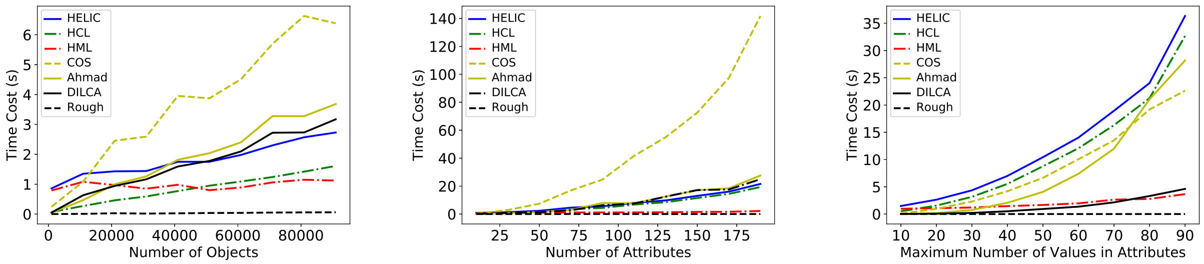
(f) Training Loss on Census Data Set.

Fig. 7. The HELIC Training Loss on Different Data Sets. The stochastic optimization method for HELIC is Adam [38], the initial learning rate is 10^{-3}, and the batch size is 20. X-axis refers to the number of iterations, and y-axis refers to the loss value of HELIC metric learning objective function Eq. (21).

We further generate synthetic data to test the computational cost of HELIC. The default data factors for synthetic data are as follows: the number of objects is $1,000$, the number of attributes is 10, and the maximum number of values in each attribute is 3. We generate three groups of data and tune one of these factors for each group. For the first group of data, the number of objects is from $1,000$ to $100,000$. For the second group of data, the number of attributes is from 10 to 200. For the third group of data, the maximum number of values in the attributes is from 10 to 100. The HELIC time cost under each data factor is shown in Fig. 8.

As shown in Fig. 8, the HELIC time cost is at the same level as most of the state-of-the-art methods. Although the Rough method has lower time complexity, it has a lower representation performance compared with others. Fig. 8(a) shows that the HELIC time cost is almost stable (from 0.84(s) to 3.47(s)), which demonstrates its good scalability w.r.t. the amount of data n_o. Actually, the small time cost increase is related to the Python built-in functions when identifying a categorical value location

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2791525, IEEE Transactions on Knowledge and Data Engineering



(a) Time Cost w.r.t. n_o.

(b) Time Cost w.r.t. n_a.

(c) Time Cost w.r.t. n_{mv}.

Fig. 8. The HELIC Time Cost w.r.t. Data Factors: Object Number n_o, Attribute Number n_a, and Maximum Number of Attribute Values n_{mv}.

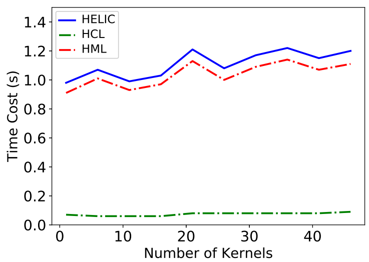


Fig. 9. The HELIC Time Cost w.r.t. Number of Kernels.

in the data. Since this cost increases by an extremely small proportion with regard to the amount of data, we can ignore it when applying HELIC. Fig. 8(b) and Fig. 8(c) demonstrate that the time cost has a quadratic relation with both n_a and n_{mv}, which is consistent with the time complexity of HELIC, as analyzed in Section 4.3. These results also show that the main HELIC cost lies in hierarchical coupling learning (HCL), while the cost of heterogeneity and metric learning (HML) is constant. The reason for this is that HELIC calculates the pairwise value relations when learning inter-attribute couplings. Hence, for categorical data with high dimension, the trade-off between capturing comprehensive complex couplings and preserving efficiency needs to be made.

To evaluate the relation between time cost and the number of kernel functions, we set the number of kernels used in HELIC from 1 to 50 and test the computational cost of HELIC on the synthetic data set with default data factors. The HELIC time cost with a different number of kernels is shown in Fig. 9. This shows that the HELIC time cost is linear to the number of kernels with a very small slope. Increasing the number of kernels only slightly affects the computational time of HELIC. This is consistent with our theoretical analysis, which indicates n_ω is linear to the time complexity of HELIC. Here, n_ω has a linear relation with the number of kernels.

5.7 Testing HELIC Stability

We evaluate HELIC stability with regard to its parameter λ, which is a parameter that controls the weight of sparsity of $\boldsymbol{\omega}$. The larger λ is, the less components are selected for the construction of the finial metric, and vice versa. Fig. 10 shows the HELIC-enabled KNN classification F-score under different settings of λ on the Krvskp data set. This illustrates that HELIC is stable for a large range of λ especially when λ is less than 1. However, the F-score drops rapidly when the value of λ increases over 10, which indicates that the regularization dominates the objective function. In this case, the learned $\boldsymbol{\omega}$ cannot well reveal the heterogeneity and may even cause to lose the learned couplings. Therefore, we suggest to choose a small value for λ. A good choice for λ is $\frac{1}{n_k}$, which decreases as the size of $\boldsymbol{\omega}$ increases.

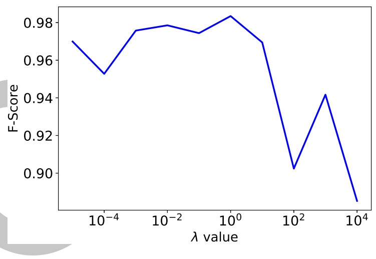


Fig. 10. The HELIC-enabled KNN Classification F-score w.r.t. λ.

6 Conclusions

Complex categorical data is often embedded with hierarchical coupling relationships and heterogeneities, which are highly challenging to model and are rarely explored. This work reports on an effective heterogeneous metric HELIC for learning hierarchical couplings within and between attributes and between attributes and classes in categorical data. HELIC analyzes the heterogeneities in hierarchical interaction spaces and integrates heterogeneous couplings in complex categorical data. Both theoretical and experimental analyses show HELIC's effectiveness and efficiency in classifying categorical data with diverse data characteristics.

The proposed method has the potential to be applied to a variety of categorical data. One noted direction in HELIC applications is to select appropriate kernels by aligning them with specific data characteristics and domain knowledge of the problems. A good selection of kernels may further improve the metric learning performance, which will be part of our further work. In addition, we are modeling more complex couplings, such as couplings between several attributes, to enhance the HELIC framework. Lastly, learning complex couplings and heterogeneities forms a challenging yet important non-IID learning problem, widely embedded in big data applications.

References

[1] D. Ienco, R. G. Pensa, and R. Meo, "From context to distance: Learning dissimilarity for categorical data clustering," *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 1, pp. 1–25, 2012.

[2] L. Ralaivola, M. Szafranski, and G. Stempfel, "Chromatic pac-bayes bounds for non-iid data: Applications to ranking and stationary β-mixing processes," *Journal of Machine Learning Research*, vol. 11, no. Jul, pp. 1927–1956, 2010.

[3] L. Cao, Y. Ou, and P. S. Yu, "Coupled behavior analysis with applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 8, pp. 1378–1392, 2012.

[4] L. Cao, "Non-iidness learning in behavioral and social data," *The Computer Journal*, vol. 57, no. 9, pp. 1358–1370, 2014.

[5] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research*, vol. 10, no. Feb, pp. 207–244, 2009.

[6] Y. Ying and P. Li, "Distance metric learning with eigenvalue optimization," *Journal of Machine Learning Research*, vol. 13, no. Jan, pp. 1–26, 2012.

[7] D. Lim, G. R. Lanckriet, and B. McFee, "Robust structural metric learning." in *ICML*, 2013, pp. 615–623.

[8] X. Gao, S. C. Hoi, Y. Zhang, J. Wan, and J. Li, "Soml: Sparse online metric learning with application to image retrieval." in *AAAI*, 2014, pp. 1206–1212.

[9] D. Lim and G. Lanckriet, "Efficient learning of mahalanobis metrics for ranking," in *ICML*, 2014, pp. 1980–1988.

[10] Y. Shi, A. Bellet, and F. Sha, "Sparse compositional metric learning," in *AAAI*, 2014, pp. 2078–2084.

[11] H. Wang, F. Nie, H. Huang, and H. Huang, "Robust distance metric learning via simultaneous l1-norm minimization and maximization." in *ICML*, 2014, pp. 1836–1844.

[12] M. Cuturi and D. Avis, "Ground metric learning," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 533–564, 2014.

[13] W. Liu, C. Mu, R. Ji, S. Ma, J. R. Smith, and S.-F. Chang, "Low-rank similarity metric learning in high dimensions," in *AAAI*, 2015, pp. 2792–2799.

[14] V. Cheng, C.-H. Li, J. T. Kwok, and C.-K. Li, "Dissimilarity learning for nominal data," *Pattern Recognition*, vol. 37, no. 7, pp. 1471–1477, 2004.

[15] Y.-M. Cheung and H. Jia, "Categorical-and-numerical-attribute data clustering based on a unified similarity metric without knowing cluster number," *Pattern Recognition*, vol. 46, no. 8, pp. 2228–2238, 2013.

[16] Y. Shi, W. Li, and F. Sha, "Metric learning for ordinal data," in *AAAI*, 2016, pp. 2030–2036.

[17] M. K. Ng, M. J. Li, J. Z. Huang, and Z. He, "On the impact of dissimilarity measure in k-modes clustering algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 503–507, 2007.

[18] F. Cao, J. Liang, D. Li, L. Bai, and C. Dang, "A dissimilarity measure for the k-modes clustering algorithm," *Knowledge-Based Systems*, vol. 26, pp. 120–127, 2012.

[19] S. Q. Le and T. B. Ho, "An association-based dissimilarity measure for categorical data," *Pattern Recognition Letters*, vol. 26, no. 16, pp. 2549–2557, 2005.

[20] A. Ahmad and L. Dey, "A method to compute distance between two categorical values of same attribute in unsupervised learning for categorical data set," *Pattern Recognition Letters*, vol. 28, no. 1, pp. 110–118, 2007.

[21] H. Jia, Y.-m. Cheung, and J. Liu, "A new distance metric for unsupervised learning of categorical data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 5, pp. 1065–1079, 2016.

[22] C. Wang, Z. She, and L. Cao, "Coupled attribute analysis on numerical data." in *AAAI*, 2013, pp. 1736–1742.

[23] C. Wang, X. Dong, F. Zhou, L. Cao, and C.-H. Chi, "Coupled attribute similarity learning on categorical data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 4, pp. 781–797, 2015.

[24] K. Zhang, Q. Wang, Z. Chen, I. Marsic, V. Kumar, G. Jiang, and J. Zhang, "From categorical to numerical: Multiple transitive distance learning and embedding," in *SDM*, 2015, pp. 46–54.

[25] D. W. Goodall, "A new similarity index based on probability," *Biometrics*, vol. 22, no. 4, pp. 882–907, 1966.

[26] S. Boriah, V. Chandola, and V. Kumar, "Similarity measures for categorical data: A comparative evaluation," in *SDM*, 2008, pp. 243–254.

[27] L. Cao, "Coupling learning of complex interactions," *Information Processing & Management*, vol. 51, no. 2, pp. 167–186, 2015.

[28] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[29] C. Stanfill and D. Waltz, "Toward memory-based reasoning," *Communications of the ACM*, vol. 29, no. 12, pp. 1213–1228, 1986.

[30] J. Xie, B. K. Szymanski, and M. J. Zaki, "Learning dissimilarities for categorical symbols," in *JMLR: Workshop on Feature Selection in Data Mining*. JMLR. org, 2010, pp. 97–106.

[31] P. Jain, B. Kulis, J. V. Davis, and I. S. Dhillon, "Metric and kernel learning using a linear transformation," *Journal of Machine Learning Research*, vol. 13, no. Mar, pp. 519–547, 2012.

[32] Y. He, W. Chen, Y. Chen, and Y. Mao, "Kernel density metric learning," in *ICDM*, 2013, pp. 271–280.

[33] S. Peng, Q. Hu, Y. Chen, and J. Dang, "Improved support vector machine algorithm for heterogeneous data," *Pattern Recognition*, vol. 48, no. 6, pp. 2072–2083, 2015.

[34] H.-J. Ye, D.-C. Zhan, and Y. Jiang, "Instance specific metric subspace learning: A bayesian approach," in *AAAI*, 2016, pp. 2272–2278.

[35] S. Jian, L. Cao, G. Pang, K. Lu, and H. Gao, "Embedding-based representation of categorical data by hierarchical value coupling learning," in *IJCAI*, 2017, pp. 1937–1943.

[36] Q. Cao, Z.-C. Guo, and Y. Ying, "Generalization bounds for metric and similarity learning," *Machine Learning*, vol. 102, no. 1, pp. 115–132, 2016.

[37] J.-P. Kahane, *Some random series of functions*. Cambridge University Press, 1993, vol. 5.

[38] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[39] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.

[40] M.-F. Balcan, A. Blum, and N. Srebro, "A theory of learning with similarity functions," *Machine Learning*, vol. 72, no. 1-2, pp. 89–112, 2008.

Chengzhang Zhu is now pursuing his PhD degree in the Faculty of Engineering and IT, University of Technology Sydney, Australia. His research interests include metric learning, non-IID learning, and data representation, in addition to general interest in data science especially machine learning.

Longbing Cao (SM06) received a PhD in pattern recognition and intelligent systems from the Chinese Academy of Science, and a PhD in computing sciences from the University of Technology. He is a professor and the Founding Director of the UTS Advanced Analytics Institute. His current research interests include data science, artificial intelligence, behavior informatics and their enterprise applications.

Qiang Liu (M14) receied PhD in computer science and technology from National University of Defense Technology (NUDT) in 2014, currently an Assistant Professor at NUDT. His research interests include machine learning, 5G network, and Internet of Things. He currently serves on the editorial review board of Artificial Intelligence Research journal.

Jianping Yin received a PhD in computer science and technology from the National University of Defense Technology (NUDT) in 1990. He is a professor at the Dongguan University of Technology. His research interests include theoretical computer science, artificial intelligence, pattern recognition and networking algorithms.

Vipin Kumar is a Regents Professor and William Norris Chair in Large-Scale Computing at UMN Twin Cities. His research interests include data mining and high-performance computing, and their application in climate, ecosystem, and biomedical domains. Kumar received a PhD in computer science from the University of Maryland. He is a Fellow of IEEE, ACM, and the American Association for the Advancement of Science.