Morphism-Based Learning for Structured Data

Kilho Shin,[1] **David Lawrence Shepard**[2]
[1]Gakushuin University, Japan, [2]UCLA Scholarly Innovation Lab., USA
yoshihiro.shin@gakushuin.ac.jp, shepard.david@gmail.com

Abstract

In mathematics, morphism is a term that indicates structure-preserving mappings between mathematical structures of the same type. Linear transformations for linear spaces, homomorphisms for algebraic structures and continuous functions for topological spaces are examples. Many data researched in machine learning, on the other hand, can include mathematical structures in them. Strings are totally ordered sets, and trees can be understood not only as graphs but also as partially ordered sets with respect to an ancestor-to-descendent order and semigroups with respect to the binary operation to determine nearest common ancestor. In this paper, we propose a generic and theoretic framework to investigate similarity of structured data through structure-preserving one-to-one partial mappings, which we call morphisms. Through morphisms, useful and important methods studied in the literature can be abstracted into common concepts, although they have been studied separately. When we study new structures of data, we will be able to extend the legacy methods for the purpose of studying the new structure, if we can define morphisms properly. Also, this view reveals hidden relations between methods known in the literature and can let us understand them more clearly. For example, we see that the center star algorithm, which was originally developed to compute sequential multiple alignments, can be abstracted so that it not only applies to data structures other than strings but also can be used to solve problems of pattern extraction. The methods that we study in this paper include edit distance, multiple alignment, pattern extraction and kernel, but it is sure that there exist much more methods that can be abstracted within our framework.

1 Introduction

Similarity of data is the most fundamental concept of machine learning. For example, clustering is to make groups of data so that members of each group are similar to one another, and classification is to predict unknown classes of data based on their similarity to known data. Thus, quantitative evaluation of similarity of data by some means is an imperative step of machine learning algorithms. In fact, various methods have been proposed in the literature to quantify similarity. For example, a *kernel* is a similarity measure, that is, the greater, the more similar (*e.g.*, (Lodhi et al. 2001; Collins and Duffy 2001)), while an *edit distance* is a dissimilarity measure, that is, the smaller, the more similar (*e.g.*, (Levenshtein 1966; Taï 1979)).

In this paper, we will propose a novel abstract approach to quantify the similarity of discrete and structured data. The key finding that drove us was the fact that plural methods of machine learning developed for data with different structures can be sometimes redefined in a common manner using the concept of *morphisms*, which abstracts differences in structures. In mathematics, morphism is a term that indicates structure-preserving mappings between mathematical structures of the same type. Linear transformation for linear spaces is a good example. In our approach, we view a datum as a set of elements that is equipped with some structure (relation) among the elements, and define morphisms as one-to-one partial set mappings that preserve the structure. On top of the concept of morphisms, we define methods to investigate similarity of data in an abstract manner. The methods in the literature that we study in this paper encompass edit distances (Levenshtein 1966; Taï 1979), sequential multiple alignments (Gusfield 1993), pattern extraction (Kao et al. 2007)) and kernels (Haussler 1999; Collins and Duffy 2001; Lodhi et al. 2001; Kashima and Koyanagi 2002; Leslie et al. 2004), but the range where our framework is effective should not be limited to them in principle.

The advantages of our theory include: (1) it bridges gaps among important methodologies of machine learning, which exist due to differences in data structures to study, and as a consequence, a method that has proven effective for a type of structures can be converted into a new method for other types of structures; For example, we unify many different definitions of edit distances for strings, trees and graphs into a single common definition; (2) Abstraction by morphisms can make a methodology developed to solve a particular problem applicable to problems of different types; For example, we show that the center star algorithm, which was developed to study sequential multiple alignments, can be used to extract structural patterns of data with various structures; (3) It provides a generic framework to engineer novel methods in a uniform manner; If we can mathematically define structures of data in the form of morphisms, abstract

methods constructed on top of morphisms can automatically apply to them.

2 A morphism-based framework

2.1 Mathematical notations

In this paper, we use the following notations.

- S and T denote arbitrary sets.
- A partial mapping μ from S to T is a subset of $S \times T$ such that, if (s,t) and (s,t') are both in μ, then $t = t'$.
- A one-to-one partial mapping μ satisfies that, if (s,t) and (s',t) are both in μ, then $s = s'$.
- The domain of μ is $\mathrm{Dom}(\mu) = \{s \mid \exists t \in T[(s,t) \in \mu]\}$.
- The range of μ is $\mathrm{Ran}(\mu) = \{t \mid \exists s \in S[(s,t) \in \mu]\}$.
- For two partial mappings $\mu \subseteq S \times U$ and $\nu \subseteq U \times T$, their composition is the partial mapping determined by $\nu \circ \mu = \{(s,t) \mid \exists u \in U[(s,u) \in \mu, (u,t) \in \nu]\} \subseteq S \times T$.
- The cardinal number of μ is denoted by $|\mu|$.
- Kronecker's delta function $\delta_{x,y}$ yields 1 if $x = y$ and 0 otherwise.
- For a propagation $(a_1, \ldots, a_n)$, $[a_1, a_{\hat{i}}, a_n]$ denotes the sub-propagation $(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$.

2.2 Introductory illustration

Our framework is based on three premises stated below:

- A datum is a collection of one or more labeled component elements;
- A similarity function to compare labels is given as prior knowledge;
- Structures among elements within data are represented by sets of one-to-one structure-preserving partial mappings between data.

We exemplify the concept with strings of alphabetic letters:

- We view a string as a collection of elements labeled with letters in an alphabet Σ through a function ℓ. If $S = a_1 \ldots a_{|S|}$ and $T = b_1, \ldots, b_{|T|}$ are two strings with $a_i \in \Sigma$ and $b_i \in \Sigma$, we view S and T as collections of labeled elements $\{s_1, \ldots, s_{|S|}\}$ and $\{t_1, \ldots, t_{|T|}\}$ with $\ell(s_i) = a_i$ and $\ell(t_i) = b_i$.
- As a similarity function to compare labels, we deploy Koronecker's delta function. That is, the similarity of elements s and t is either 0 or 1 determined by $\delta_{\ell(s),\ell(t)}$.
- We deal with the sequential orders of elements in strings through the entire set of one-to-one order-preserving partial mappings between S and T, denoted by $\mathcal{M}_{S,T}$:

$$\mathcal{M}_{S,T} = \big\{\{(s_{i_1}, t_{j_1}), \ldots, (s_{i_n}, t_{j_n})\} \in S \times T \mid n \geq 1, \\ 1 \leq i_1 < \cdots < i_n \leq |S|, 1 \leq j_1 < \cdots < j_n \leq |T|\big\}.$$

Although $\mathcal{M}_{S,T}$ does not completely determine the sequential structures of S and T, it reflects an important part of the structures. In particular, if $\mathcal{M}_{S,T}$ and the order of S is given, the order of T is uniquely determined.

2.3 Component elements, data, and morphisms

We give a formal description of our framework and introduce the *Maximum Similarity Measurement (MSM) problem*.

A space of data A datum in our framework is always a set, and $\mathscr{D}$ denotes the space of data of our interest.

Labels and a label similarity measure. In our framework, labels associated with elements are used for the purpose of evaluating the similarity among the elements. The association of elements with labels is determined through a labeling function $\ell_X : X \to \mathscr{L}$, where $\mathscr{L}$ denotes a common finite alphabet of labels. Furthermore, a label similarity function $\varphi : \mathscr{L} \times \mathscr{L} \to [0, \infty)$ is given as prior knowledge, and therefore, $\varphi(\ell_X(x), \ell_Y(y))$ can be used as a similarity measurement between elements $x \in X$ and $y \in Y$ for $X, Y \in \mathscr{D}$. The following are the axioms for φ to satisfy:

$$\varphi(\ell_1, \ell_2) \geq 0 \quad \text{(Non-negativity)}$$
$$\varphi(\ell_1, \ell_1) > 0 \quad \text{(Self-positivity)}$$
$$\varphi(\ell_1, \ell_2) = \varphi(\ell_2, \ell_1) \quad \text{(Symmetry)}$$

Furthermore, we have several desirable properties for φ to have:

$$\varphi(\ell_1, \ell_1) \geq \varphi(\ell_1, \ell_2) \quad \text{(Maximality);} \tag{1}$$
$$\varphi(\ell_1, \ell_2)\varphi(\ell_3, \ell_3) \geq \varphi(\ell_1, \ell_3)\varphi(\ell_2, \ell_3) \quad \text{(Convexity);} \tag{2}$$

and for any $n > 0, \ell_1, \ldots, \ell_n \in \mathscr{L}$ and $c_1, \ldots, c_n \in \mathbb{R}$,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j \varphi(\ell_i, \ell_j) \geq 0 \quad \text{(Positive definiteness).} \tag{3}$$

As explained later in this paper, maximality and convexity are sufficient conditions to make *morphism-based distances* pseudo-metrics, while positive definiteness (Berg, Christensen, and Ressel 1984) is a necessary condition to make *morphism-based moment kernels* positive definite. Positive definiteness of kernels is known to be necessary to take advantage of the kernel method.

When we restate problems known in the literature within our framework, the most common setting of φ is $\varphi(\ell, \ell') = \beta + (\alpha - \beta)\delta_{\ell,\ell'}$ with $\alpha > \beta \geq 0$. This φ, indeed, satisfies all of the axioms and the properties stated above. In this paper, however, we assume that φ always has non-negativity, self-positivity and symmetry and will require the others only when they are necessary.

Morphisms. As morphisms between data X and Y, we use one-to-one partial mappings from X to Y as sets that also preserve the predetermined structures of the data. Our framework provides various methods to evaluate the similarity between data, but they are commonly realized through predetermined sets of morphisms. To be specific, each pair of data $(X, Y) \in \mathscr{D} \times \mathscr{D}$ is uniquely associated with a set of morphisms, denoted by $\mathcal{M}_{X,Y}$. By allowing morphisms to be partial, we will be able to incorporate local similarities of data into evaluation of the entire similarity.

As axioms of morphisms, we require:

$id_X \in \mathcal{M}_{X,X}$ (Identity);
$\mu \in \mathcal{M}_{X,Y} \Rightarrow \mu^{-1} \in \mathcal{M}_{Y,X}$ (Inverse);
$\mu \in \mathcal{M}_{X,Y}, \nu \in \mathcal{M}_{Y,Z} \Rightarrow \nu \circ \mu \in \mathcal{M}_{X,Z}$ (Transitivity).

When we restate problems in the literature within our framework, we have examples where morphisms are not necessarily transitive. For example, the morphisms associated with the less-constrained tree edit distance are not transitive, and as a result, the triangle inequality does not hold. However, such examples are exceptional, and therefore, we include transitivity in the axioms for morphisms.

In the category theory, the transitivity is one of the axioms for a category, and $\mathscr{D}$ and morphisms described in the above becomes a category, but is not concrete, because morphisms include partial mappings.

2.4 Maximum Similarity Measurement Problem

We define a simple similarity function $\Phi_\varphi^{\mathcal{M}} : \mathscr{D} \times \mathscr{D} \to [0, \infty)$ on top of the concepts introduced in the above. We start with defining $\Phi_\varphi : \bigcup_{X,Y \in \mathscr{D}} \mathcal{M}_{X,Y} \to \mathbb{R}$ by

$$\Phi_\varphi(\mu) = \prod_{(x,y)\in\mu} \varphi(x, y). \tag{4}$$

To define the simple similarity, we can deploy a definition based on a sum of primitive similarity measurements instead of their product. In fact, for $\varphi^+ : \mathscr{L} \times \mathscr{L} \to [-\infty, \infty)$, we can define

$$\Phi^+_{\varphi^+}(\mu) = \sum_{(x,y)\in\mu} \varphi^+(x, y).$$

These definitions are, however, mutually equivalent, because we can convert $\Phi^+_{\varphi^+}$ to Φ_φ by $\varphi(x, y) = e^{\varphi^+(x,y)}$ and $\Phi_\varphi(\mu) = e^{\Phi^+\varphi^+\mu}$. In this paper, we define a similarity measure of data as follows based on Eq. (4).

Maximum Similarity Measure ment (MSM)

We define the similarity between $X, Y \in \mathscr{D}$ by

$$\Phi_\varphi^{\mathcal{M}}(X, Y) = \max\{\Phi_\varphi(\mu) \mid \mu \in \mathcal{M}_{X,Y}\}. \tag{5}$$

On top of this fundamental similarity measure, we can construct variable measures to evaluate the similarity of data from different points of view. For example, the type of problems to find morphisms μ that maximize MSM may be a target of interest of researchers, and it abstracts various concrete problems of pattern extraction. In fact, a formulation of the *morphism-based pattern extraction* (MPE) problem is introduced in Section 5. Also, to incorporate the influence of $X \setminus \mathrm{Dom}(\mu)$ and $Y \setminus \mathrm{Ran}(\mu)$ into evaluation of similarity in addition to MSM, we introduce the *morphism-based distance* (MD) in Section 3, which is an abstraction of the well-known concept of edit distances. On the other hand, the *morphism-based moment kernel* (MMK), which is introduced in Section 6, evaluates the distributions of $\Phi_\varphi(\mu)$ across morphisms $\mu \in \mathcal{M}_{X,Y}$.

2.5 A 2-MAST problem is an MSM problem

To illustrate, we see that the well known 2-MAST (Maximum Agreement Subtree) problem (Kao et al. 2007) can be formulated as a problem to find a morphism μ that maximizes $\Phi_\varphi^{\mathcal{M}}(X, Y)$ of an MSM problem.

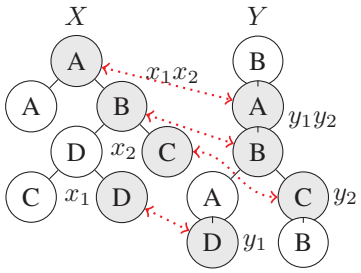

Figure 1: 2-MAST.

For this purpose, it is convenient to define rooted trees as algebraic structures: If a semigroup $(X, \cdot)$ satisfies the additional conditions of (i) $xy = yx$, (ii) $x^2 = x$ and (iii) $|\{xy, yz, zx\}| \leq 2$, it is a rooted tree by viewing X as a vertex set and xy as the nearest common ancestor of x and y in X. The root is computed by $\prod_{x\in X} x$. Furthermore, an agreement subtree S of X is an arbitrary sub-semigroup of X. Under these notations, 2-MAST problem is formulated as a problem to find two agreement subtrees $S \subseteq X$ and $T \subseteq Y$ maximum in size such that there is a semigroup isomorphism $\mu : S \to T$ that preserves vertex labels (Fig. 1). We say that S and T are mutually congruent.

With this definition, we can view 2-MAST problem as an MSM problem. We let $\mathscr{D}$ be a set of labeled rooted trees, and determine $\mathcal{M}_{X,Y}$ for $X, Y \in \mathscr{D}$ by

$$\mathcal{M}_{X,Y} = \{\mu \mid \mathrm{Dom}(\mu) \text{ and } \mathrm{Ran}(\mu) \text{ are sub-semigroups},\ \mu : \mathrm{Dom}(\mu) \to \mathrm{Ran}(\mu) \text{ is a semigroup isomorphism}\}.$$

For simplicity, we call such a morphism a *partial semigroup isomorphism*. For a label similarity function φ, on the other hand, we use $\varphi(\ell, \ell') = \alpha\delta_{\ell,\ell'}$ with $\alpha > 1$. The corresponding MSM problem requires to maximize

$$\Phi_\varphi(\mu) = \prod_{(x,y)\in\mu} \alpha\delta_{\ell_X(x),\ell_Y(y)}.$$

If $\ell_X|_{\mathrm{Dom}(\mu)} = \ell_Y \circ \mu$ holds, we have $\Phi_\varphi(\mu) = \alpha^{|\mathrm{Dom}(\mu)|} = \alpha^{|\mathrm{Ran}(\mu)|}$, and otherwise, $\Phi_\varphi(\mu) = 0$. Therefore, if μ maximizes $\Phi_\varphi(\mu)$, the pair $(\mathrm{Dom}(\mu), \mathrm{Ran}(\mu))$ determines maximum agreement subtrees.

3 Abstraction of Edit Distances

The concept of edit distances has proven to be effective to measure similarity of strings (Levenshtein 1966), trees (Taï 1979), and graphs (Neuhaus and Bunke 2007). In this section, we introduce the *morphism-based distance* within our framework as an abstraction of edit distances.

3.1 The classical notion of edit distances

Levenshtein distance for strings (Levenshtein 1966) and Taï distance for trees (Taï 1979) are well known examples of edit distances and are defined within the same framework.

An edit distance $d(X, Y)$ between two data X and Y, which can be strings and rooted trees, is determined as the minimum cost of *edit paths*. An edit path consists of one or more *edit operations*, each of which is one of (1) substituting a component y of Y for a component x of X, (2) deleting a component x of X and (3) inserting a component y of Y, and transforms X into Y. To each edit operation, a cost is assigned: we let $\psi(x, y)$, $\psi(x, \perp)$ and $\psi(\perp, y)$ denote the costs of substitution, deletion and insertion, respectively. The symbol $\perp$ denotes a gap. Then, the cost of an edit path π, denoted by $\Psi(\pi)$, is determined by the sum of the costs of the edit operations that constitute π.

On the other hand, the substitution operations specified in π determine a one-to-one partial mapping from X to Y, which is called the *trace* of π. We denote it by $\tau(\pi)$. When we let $X_\pi = X \setminus \mathrm{Dom}(\tau(\pi))$ and $Y_\pi = Y \setminus \mathrm{Ran}(\tau(\pi))$, $\Psi(\pi)$ can be restated by

$$\Psi(\pi) = \sum_{x \in X_\pi} \psi(x, \perp) + \sum_{y \in Y_\pi} \psi(\perp, y) + \sum_{(x,y) \in \tau(\pi)} \psi(x, y). \quad (6)$$

When $\Pi_{X,Y}$ denotes the entire set of edit paths from X to Y, $d(X, Y) = \min_{\pi \in \Pi_{X,Y}} \Psi(\pi)$ determines the distance.

We have a few important observations to note here.

- A trace $\tau(\pi)$ preserves the order of letters for the string case and the generation order of vertices for the rooted tree case. With mathematical terminology, these traces are structure-preserving when we view a string is a totally ordered set and a rooted tree as a partially ordered set.
- Many other edit distances including the Hamming distance for strings, the constrained distance (Zhang 1996), the less-constrained distance (Lu, Su, and Tang 2001), the degree-two distances (Zhang, Wang, and Shasha 1996) for trees, which are all variations of the Taï distance, and the graph edit distance for graphs (Neuhaus and Bunke 2007) are defined by posing certain constraints on edit paths that constitute $\Pi_{X,Y}$. Nevertheless, these edit distances can be commonly determined by $d(X, Y) = \min_{\pi \in \Pi_{X,Y}} \Psi(\pi)$ using Eq. (6) with the restricted $\Pi_{X,Y}$.
- These constraints on edit paths can be translated into additional conditions for traces to satisfy. For example, to define the graph edit distance (Neuhaus and Bunke 2007), the constraint on edit paths is that, when deleting a vertex, all the edges connected to the vertex must be deleted beforehand, and when inserting an edge, the two ends of the edge must exist. Interestingly, this elaborated constraint turns out to be equivalent to the simple condition that traces are partial graph isomorphisms.
- Constraints on edit paths for many tree edit distances are translated to constraints on traces in (Kuboyama 2007).

3.2 Morphism-based distances (MD)

In the observations above, we have learnt that the edit distances are defined in a common way regardless of the differences in $\Pi_{X,Y}$; the differences in $\Pi_{X,Y}$ reduce to differences in the associated traces; and the traces are structure preserving partial mappings.

This understanding leads us to the notion of morphism-based distances within our framework. As a preliminary, we first define cost functions $\psi(\cdot, \cdot)$ over $\mathscr{L} \times \mathscr{L}$, and $\psi(\cdot, \perp)$ and $\psi(\perp, \cdot)$ over $\mathscr{L}$ by

$$\psi(\ell, \ell') = \frac{\log \varphi(\ell, \ell) + \log \varphi(\ell', \ell')}{2} - \log \varphi(\ell, \ell');$$
$$\psi(x, \perp) = \psi(\perp, \ell) = \frac{\log \varphi(\ell, \ell)}{2} + \frac{\log c}{2}. \quad (7)$$

c is a positive constant to adjust the cost of deletion and insertion. Finally, the morphism-based distance is determined as follows:

Morphism-based Distances (MD)

For $X, Y \in \mathscr{D}$, we determine $X_\mu = X \setminus \mathrm{Dom}(\mu)$ and $Y_\mu = Y \setminus \mathrm{Ran}(\mu)$. Then, a morphism-based distance between X and Y is determined by:

$$\Psi_{\varphi,c}(\mu) = \sum_{x \in X_\mu} \psi(\ell_X(x), \perp) + \sum_{y \in Y_\mu} \psi(\perp, \ell_Y(y)) + \sum_{(x,y) \in \mu} \psi(\ell_X(x), \ell_Y(y));$$
$$d^{\mathcal{M}}_{\varphi,c}(X, Y) = \min\{\Psi_{\varphi,c}(\mu) \mid \mu \in \mathcal{M}_{X,Y}\}.$$

We should note that determining ψ is equivalent to determining φ up to positive factor of c. In fact, $c\varphi(x, y) = e^{\psi(x,\perp)+\psi(\perp,y)-\psi(x,y)}$ holds.

3.3 Conditions to be a pseudo-metric

For morphism-based distance $d^{\mathcal{M}}_{\varphi,c}$ to be a pseudo-metric, the following four axioms must be satisfied.

$d^{\mathcal{M}}_{\varphi,c}(X, Y) \geq 0$: To satisfy this axiom, we require that the values of ψ is non-negative, and therefore, we require

$$\varphi(\ell_1, \ell_2)^2 \leq \varphi(\ell_1, \ell_1)\varphi(\ell_2, \ell_2) \quad \text{and} \quad \varphi(\ell, \ell) \geq \frac{1}{c}.$$

The first requirement will be supported, if φ has maximality (Eq. (1)) or positive definiteness (Eq. (3)): If φ satisfies maximality, $\varphi(\ell_1, \ell_2) \leq \varphi(\ell_1, \ell_1)$ and $\varphi(\ell_1, \ell_2) \leq \varphi(\ell_2, \ell_2)$ imply the requirement; If φ is positive definite, the Gramian matrix $G = \begin{bmatrix} \varphi(\ell_1, \ell_1) & \varphi(\ell_1, \ell_2) \\ \varphi(\ell_2, \ell_1) & \varphi(\ell_2, \ell_2) \end{bmatrix}$ is symmetrix, and therefore, Schur decomposition yields $U^{\mathsf{T}} G U = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$ for some orthogonal matrix $U = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_2 \end{bmatrix}$; The eigenvalues λ_1 and λ_2 are non-negative, since $\lambda_k = \sum_i \sum_j u_{ik} u_{jk} \varphi(\ell_i, \ell_j) \geq 0$, and in particular, we have $\det G = \varphi(\ell_1, \ell_1)\varphi(\ell_2, \ell_2) - \varphi(\ell_1, \ell_2)^2 \geq 0$.

The second is not an excessive demand as well, because we can determine c so that $c \geq \frac{1}{\min_{\ell \in \mathscr{L}} \varphi(\ell,\ell)}$: Since $\varphi(\ell, \ell) > 0$, $\frac{1}{\min_{\ell \in \mathscr{L}} \varphi(\ell,\ell)} < \infty$ holds.

$\boldsymbol{d^{\mathcal{M}}_{\varphi,c}(X,X) = 0}$**:** By definition, $\psi(\ell,\ell) = 0$ always holds, and hence, we have $\Psi_{\varphi,c}(id_X) = 0$.

$\boldsymbol{d^{\mathcal{M}}_{\varphi,c}(X,Y) = d^{\mathcal{M}}_{\varphi,c}(Y,X)}$**:** Because φ is symmetric, we have $\Psi_{\varphi,c}(\mu) = \Psi_{\varphi,c}(\mu^{-1})$. Since $\mu^{-1} \in \mathcal{M}_{Y,X}$ by definition, $d^{\mathcal{M}}_{\varphi,c}(X,Y) \geq d^{\mathcal{M}}_{\varphi,c}(Y,X)$ holds. For the same reason, $d^{\mathcal{M}}_{\varphi,c}(X,Y) \leq d^{\mathcal{M}}_{\varphi,c}(Y,X)$ holds as well.

$\boldsymbol{d^{\mathcal{M}}_{\varphi,c}(X,Y) + d^{\mathcal{M}}_{\varphi,c}(Y,Z) \geq d^{\mathcal{M}}_{\varphi,c}(X,Z)}$**:** If φ has maximality (Eq. (1)) and convexity (Eq. (2)), the triangle inequality holds for ψ, that is, $\psi(\ell_1,\ell_2) + \psi(\ell_2,\ell_3) \geq \psi(\ell_1,\ell_3)$ and $\psi(\bot,\ell_1) + \psi(\ell_1,\ell_2) \geq \psi(\bot,\ell_2)$ hold for arbitrary $\ell_1,\ell_2,\ell_3 \in \mathscr{L}$.

Finally, we have

Theorem 1 *If $\psi(\cdot,\cdot)$, $\psi(\cdot,\bot)$ and $\psi(\bot,\cdot)$ are all non-negative and satisfy the triangle inequality, $d^{\mathcal{M}}_{\varphi,c}$ is a pseudo-metric.*

Corollary 1 *If φ satisfies maximality and convexity, $d^{\mathcal{M}}_{\varphi,c}$ is a pseudo-metric for any $c \geq \frac{1}{\min_{\ell\in\mathscr{L}} \varphi(\ell,\ell)}$.*

We cannot always expect identity of indiscernibles. In the extreme example where $\psi(\ell,\bot) = \psi(\bot,\ell) = 0$ holds for any $\ell \in \mathscr{L}$, the resulting distances are always zero. We can, however, derive a metric space from a pseudo-metric space arbitrarily given: the quotient space of a pseudo-metric space with respect to the equivalence of $x \sim y \Leftrightarrow d(x,y) = 0$ always is a metric space.

3.4 Examples

We redefine several important instances of the edit distance known in the literature as morphism-based distances within our framework. In the following descriptions, we determine data structures and morphisms individually, while we commonly assume one of the following for φ:

$$\varphi(\ell,\ell') = \begin{cases} e^2, & \ell = \ell' \\ e, & \ell \neq \ell' \end{cases} \quad \text{or} \quad \varphi(\ell,\ell') = \begin{cases} e^2, & \ell = \ell' \\ 0, & \ell \neq \ell' \end{cases}.$$

With $c = 1$, the first yields $\psi(\ell,\ell') = 1 - \delta_{\ell,\ell'}$ and $\psi(x,\bot) = \psi(\bot,y) = 1$, which is the most common setting in the literature. The second, on the other hand, yields $\psi(\ell,\ell') = \infty$ for $\ell \neq \ell'$, and therefore, refrains substitution of elements. This type of distances is known as in-del distances and is reported to show high accuracy when used with distance-based classification algorithms such as k-NN (Shin and Niiyama 2018).

Levenshtein distance (Levenshtein 1966) A datum is a totally ordered set of labeled elements, and a morphism is an arbitrary order-preserving partial mapping: We let $(S,<)$ and $(T,<)$ be finite totally ordered sets and let $\mathcal{M}_{S,T}$ consist of partial mappings μ such that if $s,s' \in \mathrm{Dom}(\mu)$ and $s < s'$, 0

Taï distance (Taï 1979) First, we assume that trees are rooted but unordered. A rooted tree is a partially ordered set, as defined as follows.

- For a finite set X, $(X,>)$ is a partially ordered set (poset).
- For any $x \in X$, the sub-poset $(X_{>x},>)$ with $X_{>x} = \{y \in X \mid y > x\}$ is totally ordered.
- There exists $\max X$, which is referred to as the root.

The order $>$ determines a generation order, and $x > y$ means that x is a proper ancestor of y. In particular, if x is the unique parent of y, we denote $x \gg y$.

A rooted ordered tree, on the other hand, is equipped with a traversal order in addition to the generation order. Intuitively speaking, a traversal order determines an order of vertices from left to right. A rooted ordered tree can be formalized by the following axioms.

- For $(X,>,\succ)$, $(X,>)$ and $(X,\succ)$ are both posets.
- For arbitrary $x,y \in X$, exactly one of $x = y$, $x > y$, $x < y$, $x \succ y$ or $x \prec y$ holds.
- For any $x \in X$, the sub-poset $(X_{>x},>)$ with $X_{>x} = \{y \in X \mid y > x\}$ is totally ordered.
- There exists $\max X$, which is referred to as the root.
- For any $x \in X$, $(X_{\ll x},\succ)$ with $X_{\ll x} = \{y \in X \mid y \ll x\}$ is totally ordered.

Taï has proved that the set of traces of edit paths for Taï distance is identical to the set of order preserving one-to-one partial mappings (Taï 1979). Hence, we determine morphisms to be one-to-one partial mappings $\mu \subseteq X \times Y$ with $x_1 > x_2 \Leftrightarrow \mu(x_1) > \mu(x_2)$ and, if X and Y are ordered, $x_1 \succ x_2 \Leftrightarrow \mu(x_1) \succ \mu(x_2)$ as well.

Less-constrained tree distance (Lu, Su, and Tang 2001) The less-constrained distance constrains edit paths of Taï distance not to include any deletion operations prior to insertion. As a result, a morphism μ is an order preserving partial mapping such that, for $\{(x_1,y_1),(x_2,y_2),(x_3,y_3)\} \subseteq \mu$,

$$x_1x_2 < x_1x_3 \Rightarrow y_1y_2 \leq y_1y_3$$

holds: $x_i < x_j$ means that x_j is a proper ancestor of x_i, and x_ix_j is the nearest common ancestor of x_i and x_j: $x_ix_j = \min\{x \in X \mid x \geq x_i, x \geq x_j\}$. The resulting morphisms are not transitive, and in fact, the distance does not support triangle inequality.

Degree-two tree distance (Zhang, Wang, and Shasha 1996) The constraint of the degree-two distance on Taï edit paths is that insertion and deletion are allowed only for vertices of degree one or two (vertices with one or two edges). To redefine the degree-two distance, we view rooted trees as semigroups, instead of partially ordered sets, so that x_ix_j is the nearest ancestor of x_i and x_j. A morphism is an arbitrary semigroup isomorphisms, but its domain or range is not necessarily a sub-semigroup.

Graph edit distance (Neuhaus and Bunke 2007) The graph edit distance requires that, when deleting a vertex, all the edges of the vertex must be deleted beforehand, and, when inserting an edge, its two ends must be included in the graph. The resulting morphisms are arbitrary partial graph isomorphisms.

3.5 Duality between MD and MSM

Theorem 2 shows an important equivalence between the problems of determining morphism-based distances and the MSM problem. We start with describing a useful lemma.

Lemma 1 *For $\mu \in \mathcal{M}_{X,Y}$, we have*

$$\Psi_{\varphi,c}(\mu) = \log \frac{(\Phi_{c\varphi}(id_X) \cdot \Phi_{c\varphi}(id_Y))^{1/2}}{\Phi_{c\varphi}(\mu)}.$$

Proof. For $X_\mu = X \setminus \mathrm{Dom}(\mu)$ and $Y_\mu = Y \setminus \mathrm{Ran}(\mu)$,

$$\begin{aligned}
&\Psi_{\varphi,c}(\mu) \\
&= \sum_{x \in X_\mu} \frac{\log \varphi(x,x) + \log c}{2} + \sum_{y \in Y_\mu} \frac{\log \varphi(y,y) + \log c}{2} \\
&\quad + \sum_{(x,y) \in \mu} \left(\frac{1}{2} \log \varphi(x,x) + \frac{1}{2} \log \varphi(y,y) - \log \varphi(x,y) \right) \\
&= \sum_{x \in X} \frac{1}{2} \log c\varphi(x,x) + \sum_{y \in Y} \frac{1}{2} \log c\varphi(y,y) - \log \Phi_{c\varphi}(\mu)
\end{aligned}$$

implies the assertion of the theorem. □

Theorem 2 is a direct corollary to Lemma 1.

Theorem 2 (Duality) *The following equality holds.*

$$d^{\mathcal{M}}_{\varphi,c}(X,Y) = \log \frac{(\Phi_{c\varphi}(id_X) \cdot \Phi_{c\varphi}(id_Y))^{1/2}}{\Phi^{\mathcal{M}}_{c\varphi}(X,Y)} \qquad (8)$$

Since the numerator of the right-hand side of Eq (8) is constant depending only on X and Y, computing a morphism-based distance $d^{\mathcal{M}}_{\varphi,c}$ is equivalent to determining $\Phi^{\mathcal{M}}_{c\varphi}$ by solving the associated MSM problem.

4 Morphism-based multiple alignments

A multiple sequence alignment (MSA) problem is defined as follows: given more than one strings $S_1, \ldots, S_n$, the problem requires an optimal multiple alignment that minimizes the value of a predetermined cost function. For example,

$$\begin{aligned}
S_1&\texttt{: -TGTAAG---} \\
S_1&\texttt{: --GC-AGGTC} \\
S_1&\texttt{: ATGC-A--T-}
\end{aligned}$$

is a multiple alignment of three strings. A popular cost function used in the literature is the sum-of-pairs cost defined by $\sum_{i=1}^{n} \sum_{j=i+1}^{n} \Psi(\pi_{ij})$: π_{ij} is the edit path that the pairwise alignment between S_i and S_j in a multiple alignment determines; Eq. (6) determines the cost $\Psi(\pi_{ij})$. In the example, we have $\Psi(\pi_{12}) = 6$ and $\Psi(\pi_{23}) = \Psi(\pi_{31}) = 5$, and therefore, the cost of the multiple alignment is 16.

We can also abstract the concept of the multiple alignment leveraging the framework of morphism-based distances.

Morphism-based Multiple Alignment (MMA)

Given $\{X_1, \ldots, X_n\} \subseteq \mathscr{D}$, a morphism-based multiple alignment problem requires to find morphisms $\mu_{ij} \in \mathcal{M}_{X_i,X_j}$ for distinct $\{i,j\} \subseteq \{1,\ldots,n\}$ that

minimize $\displaystyle\sum_{i=1}^{n} \sum_{j=i+1}^{n} \Psi_{\varphi,c}(\mu_{ij})$

subject to (1) $\mu_{ij} = \mu_{ji}^{-1}$ and (2) $\mu_{ij} \supseteq \mu_{kj} \circ \mu_{ik}$ for any distinct $\{i,j,k\} \subseteq \{1,\ldots,n\}$.

When the number of given strings is greater than two, the MSA problem is known to be NP-hard, and hence, MMA problems for $n \geq 3$ are not necessarily solvable in polynomial time. For MSA problems with $n \geq 3$, Gusfield proposed an efficient and error-bounded approximation algorithm, namely the *center star algorithm*, (Gusfield 1993). We can abstract the algorithm in a straightforward manner to solve MMA problems without loss of the original advantages of the algorithm.

Abstract Center Star Algorithm

Input: $X_1, \ldots, X_n \in \mathscr{D}$.
Output: $\mu_{ij} \in \mathcal{M}_{X_i,X_j}$ for distinct $i,j \in \{1,\ldots,n\}$ with $\mu_{ij} = \mu_{ji}^{-1}$ and $\mu_{ij} \subseteq \mu_{kj} \circ \mu_{ik}$.
Procedures:

1. For each X_i, compute morphisms $\bar{\mu}_{ij} \in \mathcal{M}_{X_i,X_j}$ for $j \in [1,\hat{i},n]$ with $d^{\mathcal{M}}_{\varphi,c}(X_i,X_j) = \Psi_{\varphi,c}(\bar{\mu}_{ij})$ and let $S_i = \sum_{\mathrm{J} \in [1,\hat{i},n]} d^{\mathcal{M}}_{\varphi,c}(X_i,X_j)$.
2. Pick $k \in \arg\min\{S_i \mid i = 1,\ldots,n\}$;
3. Determine $\mu_{ki} = \bar{\mu}_{ki}$, $\mu_{ik} = \bar{\mu}_{ki}^{-1}$ and $\mu_{ij} = \bar{\mu}_{kj} \circ \bar{\mu}_{ki}^{-1}$ for $i \neq k$ and $j \neq k$.

The computational complexity of the abstract center star algorithm is dominated by the product of n and the computational complexity of computing the morphism-based distances $d^{\mathcal{M}}_{\varphi,c}(X_i,X_j)$, and hence, if we have a polynomial time algorithm to compute the distances, the abstract center star algorithm has a polynomial time complexity.

With respect to the approximation guarantee, we can prove that Gusfield's theorem (Gusfield 1993) also holds.

Theorem 3 *We let $\{\mu_{ij}\}_{i,j}$ be a set of morphisms obtained by the abstract center star algorithm and let $\{\widehat{\mu}_{ij}\}_{i,j}$ be an optimal solution to the MMA problem. If $d^{\mathcal{M}}_{\varphi,c}$ is a pseudo metric, we have*

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} \Psi_{\varphi,c}(\mu_{ij}) \leq \left(2 - \frac{2}{n}\right) \sum_{i=1}^{n} \sum_{j=i+1}^{n} \Psi_{\varphi,c}(\widehat{\mu}_{ij}).$$

5 Abstraction of Pattern Extraction

5.1 Morphism-based pattern extraction (MPE)

The 2-MAST problem is a typical example of pattern extraction problems and is studied in the literature in a general form as the n-MAST problem, which requires to find n agreement subtrees of n rooted trees that are congruent

to one another. To abstract it within our framework, we formalize the *morphism-based pattern extraction problem* as follows.

Morphism-based Pattern Extraction (MPE)

Given $\{X_1, \ldots, X_n\} \subseteq \mathscr{D}$, a morphism-based pattern extraction problem requires to find morphisms $\mu_{ij} \in \mathcal{M}_{X_i,X_j}$ for distinct $\{i,j\} \subseteq \{1,\ldots,n\}$ that

maximize $\displaystyle\prod_{i=1}^{n}\prod_{j=i+1}^{n} \Phi_\varphi(\mu_{ij})$

subject to (1) $\mu_{ij} = \mu_{ji}^{-1}$ and (2) $\mu_{ij} = \mu_{kj} \circ \mu_{ik}$ for any distinct $\{i,j,k\} \subseteq \{1,\ldots,n\}$.

The MMA and MPE problems are almost the same except that $\mu_{ij} \subseteq \mu_{kj} \circ \mu_{ik}$ is replaced by $\mu_{ij} = \mu_{kj} \circ \mu_{ik}$. Proposition 1 clarifies the meaning of this replacement.

Proposition 1 *If morphisms μ_{ij} for distinct $\{i,j\} \subseteq \{1,\ldots,n\}$ satisfy $\mu_{ij} = \mu_{ji}^{-1}$ and $\mu_{ij} = \mu_{kj} \circ \mu_{ik}$, $\mathrm{Dom}(\mu_{ij}) = \mathrm{Dom}(\mu_{ik}) = \mathrm{Ran}(\mu_{ji}) = \mathrm{Ran}(\mu_{ki})$ holds for any distinct $\{i,j,k\} \subseteq \{1,\ldots,n\}$.*

Thus, when a solution $\{\mu_{ij}\}_{i,j}$ of a MPE problem is given, we can view $\mathrm{Dom}(\mu_{ij})$ as the extracted pattern in X_i, which is only dependent on i by Proposition 1.

We look at this idea more closely through an example with the n-MAST problem. An n-MAST problem can be viewed as an MPE problem by defining data, morphisms and a label similarity function in the same way as when we showed a 2-MAST problem is an MSM problem: $\mathscr{D}$ is a set of rooted trees defined as semigroups with respect to the nearest common ancestor operator; morphisms are partial semigroup isomorphisms; $\varphi(\ell,\ell')$ are $\alpha\delta_{\ell,\ell'}$ with $\alpha > 1$. For a solution μ_{ij}, Proposition 1 implies that we can uniquely determine $T_i = \mathrm{Dom}(\mu_{ij}) = \mathrm{Ran}(\mu_{ji})$ for each i, which is a subsemigroup, that is, a sub-tree. μ_{ij} is a congruent mapping between T_i and T_j, if it preserves labels. Furthermore, we see that $\prod_{i=1}^{n}\prod_{j=i+1}^{n} \Phi_\varphi(\mu_{ij})$ is identical to $\alpha^{(n-1)(n-2)|T_i|/2}$, if all μ_{ij} preserve labels, and to zero, otherwise.

5.2 Center star algorithm for MPE problems

Like the multiple sequence alignment problem, n-MAST problems for $n \geq 3$ are known to be NP-hard (Kao et al. 2007). Therefore, MPE problems are not necessarily solvable in polynomial time, and we need a polynomial-time approximation algorithm with a good error bound. The duality theorem 2 may inspire us to develop the algorithm based on the center star algorithm, and this idea is actually right.

Definition 1 *For $\{X, X_1, \ldots, X_n\} \subseteq \mathscr{D}$, a* pivot *around X is $(\mu_1, \ldots, \mu_n) \in \mathcal{M}_{X,X_1} \times \cdots \times \mathcal{M}_{X,X_n}$ that maximizes $S = \prod_{i=1}^{n} \Phi_\varphi(\mu_i)$ under the constraint that all of $\mathrm{Dom}(\mu_i), i = 1,\ldots,n$, are identical. The maximum value of S is called a* signature *of X.*

The following algorithm approximately solve MPE problems.

Abstract Center Star Algorithm for MPE

Input: $X_1, \ldots, X_n \in \mathscr{D}$.
Output: $\mu_{ij} \in \mathcal{M}_{X_i,X_j}$ for distinct $i,j \in \{1,\ldots,n\}$ with $\mu_{ij} = \mu_{ji}^{-1}$ and $\mu_{ij} = \mu_{kj} \circ \mu_{ik}$.
Procedures:

1. Compute a pivot $(\bar{\mu}_{i1} \ldots, \bar{\mu}_{i,\hat{i}}, \ldots, \bar{\mu}_{in})$ around each X_i and let S_i be its signature.
2. Pick $k \in \arg\max\{S_i \mid i = 1,\ldots,n\}$;
3. Determine $\mu_{ki} = \bar{\mu}_{ki}$, $\mu_{ik} = \bar{\mu}_{ki}^{-1}$ and $\mu_{ij} = \bar{\mu}_{kj} \circ \bar{\mu}_{ki}^{-1}$ for $i \neq k$ and $j \neq k$.

Theorem 4 give an error bound of the algorithm.

Theorem 4 *For $X_1, \ldots, X_n \in \mathscr{D}$, we let $\{\mu_{ij}\}_{i,j}$ be a set of morphisms obtained by the abstract center star algorithm, $\{\widehat{\mu}_{ij}\}_{i,j}$ be an optimal solution to the MPE problem. Without any loss of generality, we assume the optimal k of the abstract center star algorithm is 1 and let $\mathcal{D}$ be $\mathrm{Dom}(\mu_{1i})$. If φ is convex, we have*

$$\log\prod_{i=1}^{n}\prod_{j=i+1}^{n}\Phi_\varphi(\mu_{ij}) \geq \left(2-\frac{n}{2}\right)\log\prod_{i=1}^{n}\prod_{j=i+1}^{n}\Phi_\varphi(\widehat{\mu}_{ij}) - \frac{(n-1)(n-2)}{2}\log\prod_{x\in D}\varphi(\ell_{X_1}(x),\ell_{X_1}(x))$$

Proof. By the hypothesis, the pivot $(\bar{\mu}_{12}, \ldots, \bar{\mu}_{1,n})$ around X_1 has been used to compute μ_{ij}.

$$\prod_{i=2}^{n}\Phi_\varphi(\mu_{1i})^{n-1} = \prod_{i=2}^{n}\prod_{j=i+1}^{n}\Phi_\varphi(\mu_{1i})\Phi_\varphi(\mu_{1j})\cdot\prod_{i=2}^{n}\Phi_\varphi(\mu_{1i}) \leq \left(\prod_{x\in\mathcal{D}}\varphi(\ell_{X_1}(x),\ell_{X_1}(x))\right)^{\frac{(n-1)(n-2)}{2}}\cdot\prod_{i=1}^{n}\prod_{j=i+1}^{n}\Phi_\varphi(\mu_{ij}).$$

On the other hand, we let $\{\bar{\mu}_{ij} \mid j \in [1,\hat{i},n]\}$ be the pivot around X_i to compute the signature S_i. Then,

$$\prod_{j=2}^{n}\Phi_\varphi(\mu_{1j}) \geq \prod_{j\in[1,\hat{i},n]}\Phi_\varphi(\bar{\mu}_{ij}) \geq \prod_{j\in[1,\hat{i},n]}\Phi_\varphi(\widehat{\mu}_{ij})$$

holds, and Hence, we have

$$\prod_{i=2}^{n}\Phi_\varphi(\mu_{1i})^{n} \geq \prod_{i=1}^{n}\prod_{j\in[1,\hat{i},n]}\Phi_\varphi(\bar{\mu}_{ij}) \geq \left(\prod_{i=2}^{n}\prod_{j=i+1}^{n}\Phi_\varphi(\widehat{\mu}_{ij})\right)^{2}.$$

The assertion follows. □

6 Morphism-based Moment Kernels

For given $(X,Y) \in \mathscr{D}$, we can consider the distribution of $\Phi_\varphi(\mu)$ across $\mu \in \mathcal{M}_{X,Y}$, and the MSM problem determines the maximum. Morphism-based moment kernels, on the other hand, aim to evaluate the entire distribution.

6.1 Moments in statistics

When a real-valued random variable X associated with a probability distribution P is given, the n-th moment of X is defined by $m_n = \int_{-\infty}^{\infty} x^n P(x)dx$. It is known that these moments describe the distribution. In fact, we have that m_1 is the mean and $m_2 - m_1^2$ is the variance, but the power of moments is much more. Under some reasonable mathematical assumption, a Fourier transform $\widehat{P}(t) = \int_{-\infty}^{\infty} e^{itx} P(x)dx$ of $P(x)$ is Taylor-expanded as

$$\widehat{P}(t) = \sum_{n=0}^{\infty} \frac{t^n}{n!} \int_{-\infty}^{\infty} x^n P(x),$$

and the coefficients of degree n is identical to m_n. In other words, the series of moments uniquely determines the distribution $P(x)$. Based on this understanding, we introduce morphism-based moment kernels as follows

Morphism-based Moment Kernel (MMK)

For $X, Y \in \mathscr{D}$, we define an n-th moment kernel as

$$K_n(X,Y) = \sum_{\mu \in \mathcal{M}_{X,Y}} \Phi_\varphi(\mu)^n.$$

$K_0(X,Y)$ is $|\mathcal{M}_{X,Y}|$, while $\frac{K_1(X,Y)}{K_0(X,Y)}$ and $\frac{K_2(X,Y)}{K_0(X,Y)} - \left(\frac{K_1(X,Y)}{K_0(X,Y)}\right)^2$ yield the mean and the variance of the distribution of $\Phi_\varphi(\mu)$. For the unique determination, we have

Theorem 5 *For $X, Y \in \mathscr{D}$, if $|\Phi_\varphi(\mu) \mid \mu \in \mathcal{M}_{X,Y}\}| = n$ holds, $K_0(X,Y), \ldots, K_{n-1}(X,Y)$ uniquely determines the distribution of $\Phi_\varphi(\mu)$.*

Proof. We denote the distinct values of $\Phi_\varphi(\mu)$ by $\{x_1, \ldots, x_n\}$ and let N_i be the cardinal number of $N_i = |\{\mu \in \mathcal{M}_{X,Y} \mid \Phi_\varphi(\mu) = x_i\}|$. Then, we have

$$\begin{pmatrix} K_0(X,Y) \\ K_1(X,Y) \\ \vdots \\ K_{n-1}(X,Y) \end{pmatrix} = \begin{bmatrix} 1 & \ldots & 1 \\ x_1 & \ldots & x_n \\ \vdots & \ddots & \vdots \\ x_1^{n-1} & \ldots & x_n^{n-1} \end{bmatrix} \begin{pmatrix} N_1 \\ N_2 \\ \vdots \\ N_n \end{pmatrix}$$

$$= \mathrm{M} \begin{pmatrix} N_1 \\ N_3 \\ \vdots \\ N_n \end{pmatrix}.$$

$\det M = \prod_{i>j}(x_i - x_j) \neq 0$ implies the assertion. □

Use of kernels to analyze structured data has been intensively studied. The first important contribution in the literature was the *convolution kernel* by (Haussler 1999): For two finite sets S and T, the convolution kernel is defined by $K_C(S,T) = \sum_{(x,y)\in S\times T} k(x,y)$ and is positive definite, if $k(x,y)$ is positive definite. The convolution kernel is generalized into *mapping kernel* (Shin and Kuboyama 2008), which is in the form of $K_M(S,T) = \sum_{(x,y)\in M} k(x,y)$ for $M \subseteq S \times T$, and have shown the necessary and sufficient condition for K_M to be positive definite. From Theorem 3 of (Shin and Kuboyama 2010), Theorem 6 is derived.

Theorem 6 *If morphisms are transitive and φ is positive definite, $K_n(X,Y)$ is positive definite.*

Many kernels known in the literature can be restated as 0-th morphism-based moment kernels. The *all sequences kernel* (Shawe-Taylor and Cristianini 2004) for strings and the *elastic kernel* (Kashima and Koyanagi 2002) for trees are typical examples, but we have many more. To compute morphism-based moment kernels of higher degrees, the theory of partitionable kernels (Shin 2011) can be used.

6.2 Relation to MSM problems

Theorem 7 not only indicates the relation with MSM problems but also implies that moment kernels of too high degrees can have only the same information as $\max \Phi_\varphi(\mu)$.

Theorem 7 *If $\Phi_\varphi(\mu) > 0$ for all $\mu \in \mathcal{M}_{X,Y}$, we have*

$$\max\{\Phi_\varphi(\mu) \mid \mu \in \mathcal{M}_{X,Y}\} = \lim_{n\to\infty} K_n(X,Y)^{1/n}.$$

Proof. Because $\max_{i=1,\ldots,k} a_i = \lim_{n\to\infty} \frac{1}{n} \log \sum_{i=1}^{k} e^{na_i}$,

$$\Phi_\varphi^{\mathcal{M}}(X,Y) = \lim_{n\to\infty} \frac{1}{n} \log \sum_{\mu\in\mathcal{M}_{X,Y}} e^{n \log \Phi_\varphi(X)Y\mu}$$

$$= \lim_{n\to\infty} \log K_n(X,Y)^{1/n}.$$

holds. □

7 Future work

Although the effectiveness of our framework has been already proven through some experiments, we will run experiments in a larger scale with a wider variation of machine learning methods including but not limited to distance, multiple alignment, pattern extraction and kernel. For this purpose, we have a plan to develop utility programs that analyze an input dataset exhaustively and consistently by means of the morphism distance, the morphism-based pattern extraction and the moment kernels and others derived from appropriately parameterized pairs of $(\mathcal{M}, \varphi)$. For real application of this utility, the user will be able to select the most appropriate method based on the output of the utility and can use it for further analysis.

Acknowledgments

This work was partially supported by the Grant-in-Aid for Scientific Research (JSPS KAKENHI Grant Number 17H00762) from the Japan Society for the Promotion of Science.

References

Berg, C.; Christensen, J. P. R.; and Ressel, R. 1984. *Harmonic Analysis on semigroups. Theory of positive definite and related functions*. Springer.

Collins, M., and Duffy, N. 2001. Convolution kernels for natural language. In *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001]*, 625–632. MIT Press.

Gusfield, D. 1993. Efficient methods for multiple sequence alignment with guaranteed error bounds. *Bulletin of Mathematical Biology* 55:141–154.

Haussler, D. 1999. Convolution kernels on discrete structures. UCSC-CRL 99-10, Dept. of Computer Science, University of California at Santa Cruz.

Kao, M.-Y.; Lam, T.-W.; Sung, W.-K.; and Ting, H.-F. 2007. An even faster and more unifying algorithm for comparing trees via unbalanced bipartite matchings.

Kashima, H., and Koyanagi, T. 2002. Kernels for semi-structured data. In *the 9th International Conference on Machine Learning (ICML 2002)*, 291–298.

Kuboyama, T. 2007. *Matching and Learning in Trees*. Ph.D. Dissertation, Department of Advanced Interdisciplinary Studies, The University of Tokyo.

Leslie, C.; Eskin, E.; Cohen, A.; Weston, J.; and Noble, W. S. 2004. Mismatch string kernels for discriminative protein classification. *Bioinformatics* 20(4).

Levenshtein, V. I. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady* 10(8):707 – 710.

Lodhi, H.; Shawe-Taylor, J.; Cristianini, N.; and H., W. C. J. C. 2001. Text classification using string kernels. *Advances in Neural Information Processing Systems (NIPS 2000)* 13.

Lu, C. L.; Su, Z. Y.; and Tang, G. Y. 2001. A New Measure of Edit Distance between Labeled Trees. In *LNCS*, volume 2108, pp. 338–348. Springer-Verlag Heidelberg.

Neuhaus, M., and Bunke, H. 2007. *Bridging the gap between graph edit distance and kernel machines*. World Scientific.

Shawe-Taylor, J., and Cristianini, N. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Shin, K., and Kuboyama, T. 2008. A generalization of Haussler's convolution kernel - mapping kernel. In *ICML 2008*.

Shin, K., and Kuboyama, T. 2010. A generalization of haussler's convolution kernel - mapping kernel and its application to tree kernels. *J. Comput. Sci. Technol* 25(5):1040–1054.

Shin, K., and Niiyama, T. 2018. Parameterized mapping distances for semi-structured data. In *ICAART 2018 (Revised Selected Papers), LNCS 11352*, 443–466.

Shin, K. 2011. Partitionable kernels for mapping kernels. In *ICDM 2011*, 645–654.

Taï, K. C. 1979. The tree-to-tree correction problem. *journal of the ACM* 26(3):422–433.

Zhang, K.; Wang, J. T. L.; and Shasha, D. 1996. On the editing distance between undirected acyclic graphs. *International Journal of Foundations of Computer Science* 7(1):43–58.

Zhang, K. 1996. A Constrained Edit Distance Between Unordered Labeled Trees. *Algorithmica* 15:205–222.