A Generalization of Haussler's Convolution Kernel — Mapping Kernel

Kilho Shin YSHIN@CMUJ.JP
Carnegie Mellon CyLab Japan, 3-3 Higashi-kawasaki-cho, Chuo-ku, Kobe, Hyogo, Japan

Tetsuji Kuboyama KUBOYAMA@GAKUSHUIN.AC.JP
Gakushuin University, Mejiro, Toshima-ku, Tokyo, Japan

Abstract

Haussler's convolution kernel provides a successful framework for engineering new positive semidefinite kernels, and has been applied to a wide range of data types and applications. In the framework, each data object represents a finite set of finer grained components. Then, Haussler's convolution kernel takes a pair of data objects as input, and returns the sum of the return values of the predetermined primitive positive semidefinite kernel calculated for all the possible pairs of the components of the input data objects. On the other hand, the *mapping kernel* that we introduce in this paper is a natural generalization of Haussler's convolution kernel, in that the input to the primitive kernel moves over a predetermined subset rather than the entire cross product. Although we have plural instances of the mapping kernel in the literature, their positive semidefiniteness was investigated in case-by-case manners, and worse yet, was sometimes incorrectly concluded. In fact, there exists a simple and easily checkable necessary and sufficient condition, which is generic in the sense that it enables us to investigate the positive semidefiniteness of an arbitrary instance of the mapping kernel. This is the first paper that presents and proves the validity of the condition. In addition, we introduce two important instances of the mapping kernel, which we refer to as the *size-of-index-structure-distribution* kernel and the *edit-cost-distribution* kernel. Both of them are naturally derived from well known (dis)similarity measurements in the literature (*e.g.* the maximum agreement tree, the edit distance), and are reasonably expected to improve the performance of the existing measures by evaluating their distributional features rather than their peak (maximum/minimum) features.

1. Introduction

Haussler's convolution kernel (Haussler, 1999) has been used as a general framework to tailor known primitive kernels to the context of specific applications. In this section, we first review a degenerated form of Haussler's convolution kernel, which proves in fact to be equivalent to the general form of Haussler's convolution kernel (see 2.2). Let each data point x in a space χ be associated with a finite subset χ'_x of a common space χ'. Furthermore, we assume that a kernel $k : \chi' \times \chi' \to \mathbb{R}$ is given. Then, Haussler's convolution kernel $K : \chi \times \chi \to \mathbb{R}$ is defined as follows (see 2.2).

$$K(x, y) = \sum_{(x',y') \in \chi'_x \times \chi'_y} k(x', y') \tag{1}$$

Haussler proved that, if $k(x', y')$ is positive semidefinite, then so is $K(x, y)$. Haussler's convolution kernel is known to have a wide range of application (Lodhi et al., 2001; Collins & Duffy, 2001; Suzuki et al., 2004).

On the other hand, the *mapping kernel* is a natural generalization of Haussler's convolution kernel, and is defined by Eq. (2) for $\{M_{x,y} \subseteqq \chi'_x \times \chi'_y \mid (x, y) \in \chi^2\}$. The problem that the present paper addresses is to determine whether the mapping kernel is positive semidefinite.

$$K(x, y) = \sum_{(x',y') \in M_{x,y}} k(x', y') \tag{2}$$

The main contribution of the present paper is to present a necessary and sufficient condition for the mapping kernel $K(x, y)$ defined by Eq. (2) to be positive semidefinite for all possible choices of positive semidefinite $k(x', y')$. More specifically, we prove that the condition is that the *mapping system* $\{M_{x,y} \mid (x, y) \in \chi^2\}$ is *transitive*, *i.e.*, $(x', y') \in M_{x,y} \wedge (y', z') \in M_{y,z} \Rightarrow (x', z') \in M_{x,z}$. Haussler's convolution kernel is indeed the special case of the mapping kernel for $\{M_{x,y} = \chi'_x \times \chi'_y\}$, which is apparently transitive.

We see plural instances of the mapping kernel in the literature, and some of them were mistreated in respective manners.

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008. Copyright 2008 by the author(s)/owner(s).

- Although the *elastic tree kernel* (Kashima & Koyanagi, 2002) was treated as an instance of Haussler's convolution kernel, it is, in fact, an instance of the mapping kernel. Therefore, the positive semidefiniteness of the kernel should not have been determined based on Haussler's theorem.
- The *codon-improved kernel* (Zien et al., 2000) was claimed to be unconditionally positive semidefinite, since it was viewed as an instance of the polynomial kernel. The kernel, in fact, is an instance of the mapping kernel under certain settings of weights.

That is to say, the positive semidefiniteness of the aforementioned kernels were concluded on wrong grounds, and in fact, the conclusion regarding the codon-improved kernel is wrong — in reality, it is not necessarily positive semidefinite.

The kernels introduced in (Menchetti et al., 2005) and (Kuboyama et al., 2006) are also instances of the mapping kernel. In contrast to the elastic and codon-improved kernels, their positive semidefiniteness was properly investigated, albeit in specific manners.

This is the first paper that recognizes the mapping kernel as a generic class of kernels, and presents a necessary and sufficient condition that a mapping kernel becomes positive semidefinite. Furthermore, the condition is simple, intuitive and easy to check, and therefore, would make engineering of new instances of the mapping kernel easier, more efficient and more effective to a large extent.

As the second contribution of the present paper, we take advantage of the mapping kernel, and present a way to augment a couple of well-known frameworks to engineer similarity functions for discretely structured objects (*e.g.* strings, trees, general graphs).

It is known that the *maximum* sizes of shared substructures of the objects can be used as a good measure of similarities of the objects. The *maximum agreement subtree* is a good example. Also, the *edit distance* has been applied to various types of objects. An edit distance between two data objects is generally defined as the minimum cost of edit scripts that transform one object into the other.

These two frameworks are common in that they only focus on the maximum/minimum values of the similarity measures (*i.e.* the sizes of shared substructures and the costs of edit scripts), and therefore, only those substructures with the maximum sizes or those edit scripts with the minimum costs can contribute to the similarity functions. It is, however, reasonably presumable that distributional features of the measurements may carry useful information with regard to similarities of objects, and more accurate similarity functions can be engineered by evaluating the distributional features.

Based on the aforementioned consideration, we introduce two novel classes of kernels (similarity functions) each evaluating the distributional features of the sizes of shared substructures or the costs of edit scripts. Also, we show a general way to view them as mapping kernels. By virtue of our simple criteria for positive semidefinite mapping kernels, we can easily determine whether instances of the new kernel classes are positive semidefinite, and, if they are, we can take advantage of sophisticated classifiers such as support vector machines (SVM). In 3.1 and 3.2, we see that the examples of distribution-based similarity functions derived from maximum agreement subtrees and general tree edit distances are positive semidefinite, while those derived from *maximum refinement trees* (Hein et al., 1996) and *less-constrained tree edit distance* (Lu et al., 2001) are not.

2. The Mapping Kernel

In this section, as a preliminary, we quickly review the positive semidefinite kernel (2.1) and Haussler's convolution kernel (2.2). Then, we describe our main theorem with regard to the mapping kernel (2.3).

2.1. The Positive Semidefinite Kernel

A kernel $K : \chi \times \chi \longrightarrow \mathbb{R}$ is said to be positive semidefinite, if, and only if, for arbitrary $x_1, \ldots, x_n \in \chi$, the corresponding Gram matrix $G = [K(x_i, x_j)]_{i,j=1,\ldots,n}$ is a positive semidefinite matrix. Positive semidefiniteness of kernels is a critical condition for reproducing kernel Hilbert spaces to exist. In simpler cases where a data point space χ is finite, this condition is equivalent to the property that there exists a mapping $\Phi : \chi \longrightarrow \mathbb{R}^N$ such that $K(x, y) = \Phi(x)\Phi(y)^{\mathsf{T}}$.

In this paper, by a positive semidefinite matrix, we mean a real symmetric matrix (*i.e.* $A^{\mathsf{T}} = A$) that satisfies one of, hence, all of the mutually equivalent conditions stated below, where $\dim A = n$.

- $(c_1, \ldots, c_n)A(c_1, \ldots, c_n)^{\mathsf{T}} \geq 0$ for $\forall (c_1, \ldots, c_n) \in \mathbb{R}^n$.
- A has only non-negative real eigenvalues.
- There exists an n-dimensional orthogonal matrix P (*i.e.* $P^{\mathsf{T}}P = E_n$) such that $P^{\mathsf{T}}AP$ is a diagonal matrix with non-negative elements.
- $A = B^{\mathsf{T}}B$ for some $m \times n$ real matrix B.

2.2. Haussler's Convolution Kernel

Hausler's theorem (Haussler, 1999, Theorem 1) asserts the positive semidefiniteness of Haussler's R-convolution kernel, and Theorem 1 presents its special case for $D = 1$.

Theorem 1. *Let $k : \chi' \times \chi' \to \mathbb{R}$ be a positive semidefinite kernel. Given a relation $R \subseteq \chi' \times \chi$, $K : \chi \times \chi \to \mathbb{R}$ defined by Eq. (3) is also positive semidefinite.*

$$K(x, y) = \sum_{(x',x)\in R} \sum_{(y',y)\in R} k(x', y') \tag{3}$$

It is interesting to note that Haussler's theorem for $D > 1$ is obtained as a corollary to Theorem 1.

Corollary 1. (Haussler, 1999) *Let $k_d : \chi'_d \times \chi'_d \longrightarrow \mathbb{R}$ be positive semidefinite kernels for $d = 1, \ldots, D$. Given a relation $R \subset \chi'_1 \times \cdots \times \chi'_D \times \chi$, the kernel $K : \chi \times \chi \longrightarrow \mathbb{R}$ defined below is also positive semidefinite.*

$$K(x, y) = \sum_{(x'_1,\ldots,x'_D,x)\in R} \sum_{(y'_1,\ldots,y'_D,y)\in R} \prod_{d=1}^{D} k_d(x'_d, y'_d)$$

2.3. Definition and Main Theorem

Letting χ'_x denote $\{x' \in \chi' \mid (x', x) \in R\}$, Eq. (1) gives an equivalent form of Eq. (3). On the other hand, the *mapping kernel* is defined so that (x', y') moves over a subset $M_{x,y}$ of $\chi'_x \times \chi'_y$ rather than the entire cross product $\chi'_x \times \chi'_y$ (Eq. (2)).

The present paper shows that the mapping kernel is positive semidefinite for all possible choices of positive semidefinite underlying kernels k, if, and only if, $\{M_{x,y} \mid x, y \in \chi\}$ is *transitive* (Definition 2).

Therefore, for an arbitrary non-transitive $\{M_{x,y}\}$, a positive semidefinite underlying kernel $k(x', y')$ exists such that the resulting $K(x, y)$ is not positive semidefinite (4.1.2). On the other hand, $K(x, y)$ may be positive semidefinite even for a non-transitive $\{M_{x,y}\}$ and a positive semidefinite $k(x', y')$ (Example 1).

Example 1. The (k, m)-mismatch kernel $K_{(k,m)}(x, y)$ is positive semidefinite (Leslie et al., 2004). When χ'_x and χ'_y denote the sets of k-mers in x and y, $K_{(k,m)}(x, y)$ can be regarded as a mapping kernel for the non-transitive $\{M_{x,y}\}$ defined as follows.

$$M_{x,y} = \{(x', y') \mid K_{(k,m)}(x', y') \neq 0\} \subseteq \chi'_x \times \chi'_y$$

$$K_{(k,m)}(x, y) = \sum_{(x',y')\in M_{x,y}} K_{(k,m)}(x', y')$$

The result is formalized as follows.

Definition 1. A *mapping system* $\mathcal{M}$ is a triplet $(\chi, \{\chi'_x \mid x \in \chi\}, \{M_{x,y} \subseteq \chi'_x \times \chi'_y \mid (x, y) \in \chi^2\})$ such that $|M_{x,y}| < \infty$ and $(y', x') \in M_{y,x}$ if $(x', y') \in M_{x,y}$.

Definition 2. A mapping system $(\chi, \{\chi'_x\}, \{M_{x,y}\})$ is said to be *transitive*, if, and only if, $(x'_1, x'_2) \in M_{x_1,x_2} \wedge (x'_2, x'_3) \in M_{x_2,x_3} \Rightarrow (x'_1, x'_3) \in M_{x_1,x_3}$ holds for arbitrary $x_i \in \chi$ and $x'_i \in \chi'_{x_i}$ $(i = 1, 2, 3)$.

Definition 3. An *evaluating system* $\mathcal{E}$ for a mapping system $(\chi, \{\chi'_x\}, \{M_{x,y}\})$ is a triplet $(\chi', k, \{\gamma_x \mid x \in \chi\})$ with a positive semidefinite *underlying kernel* $k : \chi' \times \chi' \to \mathbb{R}$ and *projections* $\gamma_x : \chi'_x \to \chi'$.

Definition 4. For a mapping system $\mathcal{M} = (\chi, \{\chi'_x\}, \{M_{x,y}\})$ and an evaluating system $\mathcal{E} = (\chi', k, \{\gamma_x\})$ for $\mathcal{M}$, the *mapping kernel* with respect to $\mathcal{M}$ and $\mathcal{E}$ is defined by Eq. (4).

$$K(x, y) = \sum_{(x',y')\in M_{x,y}} k(\gamma_x(x'), \gamma_y(y')) \tag{4}$$

Now, our main theorem is described as follows, and its proof is given in Section 4.

Theorem 2. *For a mapping system $\mathcal{M}$, the following are equivalent to each other.*

1. *$\mathcal{M}$ is transitive.*
2. *For an arbitrary evaluating system $\mathcal{E}$ for $\mathcal{M}$, the mapping kernel with respect to $\mathcal{M}$ and $\mathcal{E}$ is positive semidefinite.*

It is possible to prove (1) $\Rightarrow$ (2) of Theorem 2 as a corollary to Theorem 1. Nevertheless, our direction in the present paper is opposite — we like to view Theorem 1 as a trivial corollary to Theorem 2. In fact, we will prove Theorem 2 without assuming Theorem 1 in Section 4.

3. Similarity Functions Based on Distributions

In this section, we introduce two new classes of the mapping kernel. The kernels are expected to improve the classification performance of known similarity measurements by evaluating their distributional features.

3.1. Size-of-index-structure-distribution Kernels

When some structures are commonly derived from two data objects, the structures may carry information with regard to similarities between the data objects. In this paper, we call such structures *index structures*.

The *agreement subtree* is a good example of the index structure, when data objects are represented as

trees. An agreement subtree between plural input trees is usually defined as a subtree *homeomorphically* included in all the input trees (Berry & Nicolas, 2004). In the present paper, we assume that the input trees are a pair of trees. Even when we fix the input tree pair, there may exist more than one agreement subtree, and the maximum size of the agreement subtrees can be naturally viewed as a measure of similarities between the input trees. The *maximum agreement subtrees* (MAST) problem is the problem to determine at least one agreement subtree with the *maximum size* among the possible agreement subtrees for the input trees. The MAST problem has been extensively studied from the application point of view (*e.g.* evolutionary trees (Hein et al., 1996; Berry & Nicolas, 2004), shape-axis trees (Pelillo, 2002)) as well as from the algorithm efficiency point of view.

When using the size of the maximum agreement subtrees as a similarity measurement between trees, we discard those agreement subtrees smaller in size than the maximum ones, and therefore, they do not contribute to the final evaluation at all. It is, however, reasonable to think that distributional features of the sizes of agreement subtrees may carry useful information with regard to similarities of the trees.

Based on the aforesaid consideration, we introduce the kernel of Eq. (5), which evaluates distributional features of the sizes of agreement subtrees. In Eq. (5), we let $\mathsf{AST}(x, y)$ denote the set of the agreement subtrees between x and y, and $f : \mathbb{N} \to \mathbb{R}_+ = \{y \geq 0 \mid y \in \mathbb{R}\}$ be an increasing function.

$$K(x, y) = \sum_{t \in \mathsf{AST}(x,y)} f(\mathsf{size\_of}(t)) \tag{5}$$

If x and y are rooted trees of bounded degree, and if $f(n) = \alpha^n$ or $f(n) = n$, for example, there exist polynomial-time efficient algorithms to calculate $K(x, y)$.

Beside the advantages due to the distributional features, the kernel could provide the advantage of using sophisticated classifiers such as SVM (Cristianini & Shawe-Taylor, 2000). In fact, our contribution asserts that $K(x, y)$ is positive semidefinite as follows. First, $K(x, y)$ can be viewed as a mapping kernel under the following notation.

- χ'_x is the set of the subtrees of x.
- $M_{x,y} = \{(x', y') \in \chi'_x \times \chi'_y \mid x' \cong y'\}$, where $x' \cong y'$ means that they are homeomorphic as trees.
- $k(x', y') = \begin{cases} f(\mathsf{size\_of}(x')) & \text{if } \mathsf{size\_of}(x') = \mathsf{size\_of}(y') \\ 0 & \text{otherwise.} \end{cases}$

It is easy to see that $\{M_{x,y}\}$ is transitive and $k(x', y')$ is positive semidefinite. Hence,

$$K(x, y) = \sum_{t \in \mathsf{AST}(x,y)} f(\mathsf{size\_of}(t)) = \sum_{(x', y') \in M_{x,y}} k(x', y')$$

is positive semidefinite by Theorem 2.

Besides the maximum agreement subtree, the *maximum refinement subtree* (Hein et al., 1996; Berry & Nicolas, 2004), *maximum subtree isomorphism* (Pelillo, 2002; Aoki et al., 2003) and *maximum agreement supertree* (Jansson et al., 2005) are also used as index structures for trees. As for general graphs, the *maximal common clique* included in an input pair of graphs is also studied in association with MAST in (Pelillo, 2002).

For each of those index structures, we can define kernels in the same way as for MAST. We have only to replace $\mathsf{AST}(x, y)$ in Eq. (5) with the set of the respective index structures. Moreover, except for the maximum refinement subtree, through the same discussion as for MAST, the kernels prove to be positive semidefinite.

Interestingly, Theorem 2 also implies that the kernels defined based on the minimum refinement subtree are not necessarily positive semidefinite. The minimum refinement subtree for $x' \subseteqq x$ and $y' \subseteqq y$ is defined as the minimum tree t such that both x' and y' can be derived from t through a sequence of *edge contractions*, and the maximum refinement subtree problem (*a.k.a.* the maximum compatible tree problem) is the problem to find a minimum refinement subtree with the largest size. Different from the agreement subtree, the relation of having a refinement is not an equivalence relation — even if x' and y', and y' and z', have refinement subtrees, x' and z' do not necessarily have a refinement subtree. This implies that the corresponding $M_{x,y}$ is not necessarily transitive. Therefore, Theorem 2 asserts that the corresponding $K(x, y)$ is not necessarily positive semidefinite.

3.2. Edit-cost-distribution Kernels

The *Edit distance* is also used as an effective measure of similarities between discrete data structures (*e.g.* (Wagner & Fischer, 1974) for strings, (Barnard et al., 1995) for trees, (Bunke, 1997) for general graphs).

Let x be an object consisting of one or more components. For example, a string consists of one or more characters which are laid out on a line. For another example, a graph consists of one or more vertices and edges, and each edge connects a vertex to another. We first give a general definition of *edit operations*, *edit*

scripts, *edit costs* and *edit distances* for such objects.

An edit operation is an operation on a component of x, and is one of (i) substituting a component b for a component a of x (denoted by $\langle a \to b\rangle$), (ii) deleting a component a of x (denoted by $\langle a \to \bullet\rangle$), and (iii) inserting a component a into x (denoted by $\langle \bullet \to a\rangle$). An edit script is a sequence of zero or more edit operations which transforms an object into another. When a cost $\gamma\langle a \to b\rangle \in \mathbb{R}$ is given for each edit operation $\langle a \to b\rangle$[1], the cost $\gamma(\sigma)$ of an edit script σ is the sum of the costs of the edit operations that comprise σ. Finally, an edit distance $d(x, y)$ between objects x and y is defined by:

$$d(x, y) = \min\{\gamma(\sigma) \mid \sigma \text{ transforms } x \text{ into } y\}.$$

Therefore, those edit scripts with larger costs than the minimum cost do not contribute to the final edit distance. In contrast, by introducing kernels by Eq. (6) with a decreasing function $f : \mathbb{R}_+ \to \mathbb{R}_+$, we try to take advantage of the information that those discarded edit scripts potentially carry.

$$K(x, y) = \sum_{\sigma \text{ transforms } x \text{ into } y} f(\gamma(\sigma)) \qquad (6)$$

It is important to note that there exists a natural interpretation of Eq. (6). In a natural setting where the cost $\gamma\langle a \to b\rangle$ is defined as the negative logarithm of the probability that the substitution of b for a (a or b could be $\bullet$) would occur (*e.g.* (Li & Jiang, 2005; Salzberg, 1997)), we let $f(x) = e^{-x}$. For an edit script $\sigma = \langle x'_1 \to y'_1\rangle \cdots \langle x'_n \to y'_n\rangle$ transforming x into y, $f(\gamma(\sigma))$ is evaluated as follows.

$$\begin{aligned} f(\gamma(\sigma)) &= e^{-\gamma(\sigma)} = e^{-\sum_{i=1}^{n} \gamma\langle x'_i \to y'_i\rangle} \\ &= e^{-\sum_{i=1}^{n} -\log \Pr(x'_i \to y'_i)} = \prod_{i=1}^{n} \Pr(x'_i \to y'_i) \end{aligned}$$

Hence, $K(x, y)$ by Eq. (6) equals the total probability that x would be transformed into y.

Usage of sophisticated classifiers such as SVM is another potential advantage of the kernels of the form of Eq. (6). In fact, as shown below, the kernels can be viewed as mapping kernels, if we can pose the following four assumptions.

[1]Usually, components are labeled with elements of an alphabet, and costs of edit operations are defined on the labels rather than on the components. However, for simplicity, we assume that the cost function is defined over the space of objects in the present paper. In addition, to make the resulting edit distance be a distance metric, the costs are often assumed to be a distance metric.

1. The cost function is symmetric (*i.e.* $\gamma\langle a \to b\rangle = \gamma\langle b \to a\rangle$).
2. We let $f(x) = e^{-cx}$ for some positive constant c.
3. In order to avoid calculating infinite sums, we take only *irreducible* edit scripts into consideration in calculating Eq. (6) — Assume that $\sigma = \langle x'_1 \to y'_1\rangle \ldots \langle x'_n \to y'_n\rangle$ transforms x into y. σ is irreducible, if, and only if, (1) x'_i (resp. y'_i) is either a component of x (resp. y) or $\bullet$ and (2) exactly one edit operation $\langle x'_i \to y'_i\rangle$ is applied to each component of x and y.
4. If two irreducible edit scripts differ from each other only in the order of the included edit operations, they are identified in calculating Eq. (6), that is, they are evaluated only once.

For $\sigma = \langle x'_1 \to y'_1\rangle \ldots \langle x'_n \to y'_n\rangle$, we assume that x'_i and y'_i are respectively components of x and y, if, and only if, $i \in \{1, \ldots, m(\sigma)\}$, and call $\langle x'_1 \to y'_1\rangle \cdots \langle x'_{m(\sigma)} \to y'_{m(\sigma)}\rangle$ the *core* of σ. Then, $\gamma(\sigma)$ and $K(x, y)$ are evaluated as follows.

$$\begin{aligned} \gamma(\sigma) &= \sum_{i=1}^{m(\sigma)} (\gamma\langle x'_i \to y'_i\rangle - \gamma\langle x'_i \to \bullet\rangle - \gamma\langle \bullet \to y'_i\rangle) \\ &\quad + \sum_{x' \in x} \gamma\langle x' \to \bullet\rangle + \sum_{y' \in y} \gamma\langle \bullet \to y'\rangle \end{aligned}$$

$$\begin{aligned} K(x, y) = &\prod_{\xi \in x} f(\gamma\langle \xi \to \bullet\rangle) \cdot \prod_{\eta \in y} f(\gamma\langle \bullet \to \eta\rangle) \cdot \\ &\left[\sum_{\sigma} \left(\prod_{i=1}^{m(\sigma)} \frac{f(\gamma\langle x'_i \to y'_i\rangle)}{f(\gamma\langle x'_i \to \bullet\rangle) f(\gamma\langle \bullet \to y'_i\rangle)} \right)\right] \end{aligned} \qquad (7)$$

In Eq. (7), the first two factors of the right-hand side are functions of x and y, and therefore, we denote them by $g(x)$ and $g(y)$, respectively. On the other hand, the last factor is a function of $x' = (x'_1, \ldots, x'_{m(\sigma)})$ and $y' = (y'_1, \ldots, y'_{m(\sigma)})$, and is denoted by $k(x', y')$. We define $M_{x,y}$ as follows.

$$\begin{aligned} M_{x,y} = \{&((x'_1, \ldots, x'_m), (y'_1, \ldots, y'_m)) \mid \\ &\exists\sigma[\langle x'_1 \to y'_1\rangle \cdots \langle x'_m \to y'_m\rangle \text{ is the core of } \sigma]\} \end{aligned}$$

Then, the following holds

$$\begin{aligned} K(x, y) &= g(x) \cdot g(y) \cdot \left(\sum_{(x', y') \in M_{x,y}} k(x', y') \right) \\ &= g(x) \cdot g(y) \cdot \bar{K}(x, y) \end{aligned}$$

In particular, $\bar{K}(x, y)$ is a mapping kernel, and $K(x, y)$ is positive semidefinite, if, and only if, so is $\bar{K}(x, y)$,

since $g(x)$ cannot take the value 0. The kernel $\bar{K}(x, y)$, however, is not necessarily positive semidefinite, even if $k(x', y')$ is positive semidefinite, since $\{M_{x,y}\}$ is not necessarily transitive. We will investigate this problem taking the *tree edit distance* as an example.

For the tree edit distance, the edit operations act on vertices of trees. For a pair (x', y') to be the core of some irreducible tree edit script, it is necessary and sufficient that φ defined by $\varphi(x'_i) = y'_i$ preserves the ancestor-descendent relation and the sibling (left-to-right) relation (Tai, 1979). Therefore, $M_{x,y}$ for the general tree edit distance is defined as follows, where $x'_i < x'_j$ means x'_j is an ancestor of x'_i and $x'_i \prec x'_j$ means x'_j is located on the right side of x'_i.

$$M_{x,y} = \{((x'_1, \ldots, x'_m), (y'_1, \ldots, y'_m)) \mid [x'_i < x'_j \Leftrightarrow y'_i < y'_j] \wedge [x'_i \prec x'_j \Leftrightarrow y'_i \prec y'_j]\} \quad (8)$$

It is straightforward to verify that $\{M_{x,y}\}$ is transitive. Therefore, Theorem 2 asserts that, if $k(x', y')$ is positive semidefinite, so is $\bar{K}(x, y)$ for this $\{M_{x,y}\}$.

On the other hand, two subclasses of the general tree edit distance have been proposed. They are *constrained* (*a.k.a.* structure-preserving) tree edit distance (Zhang, 1995) and *less-constrained* (*a.k.a.* alignable) tree edit distance (Lu et al., 2001).

Those subclasses of the general tree edit distance determine respective $M_{x,y}$, which are generally proper subsets of those define by (8). Since $\{M_{x,y}\}$ for the constrained tree edit distance is easily verified to be transitive, the resulting $\bar{K}(x, y)$ turns out positive semidefinite by virtue of Theorem 2. In contrast to the constrained edit distance, $\{M_{x,y}\}$ for the less-constrained tree edit distance is not transitive. Therefore, Theorem 2 implies that $\bar{K}(x, y)$ is not necessarily positive semidefinite.

4. Proof of Theorem 2

4.1. Key Lemma

Let X^{ij} be m-dimensional square matrices parameterized by $(i, j) = \{1, \ldots, n\}^2$, and let X denote the derived mn-dimensional square matrix $[X^{ij}]_{i,j=1,\ldots,n}$ — the $(m(i-1)+k, m(j-1)+l)$-element of X, denoted by X^{ij}_{kl}, is defined to be the (k, l)-element of X^{ij}.

Furthermore, for an m-dimensional square matrix A, $\mathrm{smry}_A(X)$ denotes the n-dimensional square matrix $[\mathrm{tr}(A^\mathsf{T} X^{ij})]_{i,j=1,\ldots,n}$. Note that the (i, j)-element of $\mathrm{smry}_A(X)$ is given by Eq. (9).

$$\mathrm{tr}(A^\mathsf{T} X^{ij}) = \sum_{k=1}^{m}\sum_{l=1}^{m} A_{kl} X^{ij}_{kl} \quad (9)$$

Proposition 1. *For an m-dimensional square matrix A, the following are equivalent to each other.*

1. *A is positive semidefinite.*
2. *$\mathrm{smry}_A(X)$ is positive semidefinite for an arbitrary mn-dimensional positive semidefinite matrix X.*

Proof. First, we prove the assertion assuming that A is diagonal, whose I-th diagonal element is α_I.

The condition 2 implies 1, since we see $\alpha_I \geq 0$ for any I by letting X be the sparse matrix such that X_{kl} is 1, if $k = l = I$, and 0, otherwise.

On the other hand, the converse follows from Eq. (10), since $\mathrm{smry}_A(X) = Z^\mathsf{T} Z$ holds for the $m^2 n \times n$ matrix Z such that $Z_{mn(I-1)+m(k-1)+i,j} = \sqrt{\alpha_I} Y^{kj}_{iI}$, where Y is an mn-dimensional matrix such that $X = Y^\mathsf{T} Y$.

$$\begin{aligned} \mathrm{tr} A^\mathsf{T} X^{ij} &= \sum_{I=1}^{m} \alpha_I \left(\sum_{k=1}^{n}\sum_{l=1}^{m} Y^{ki}_{lI} Y^{kj}_{lI} \right) \quad (10) \\ &= \sum_{I=1}^{m}\sum_{k=1}^{n}\sum_{l=1}^{m} (\sqrt{\alpha_I} Y^{ki}_{lI})(\sqrt{\alpha_I} Y^{kj}_{lI}) \end{aligned}$$

The general cases for non-diagonal A reduces to the diagonal case, since, for P such that $P^\mathsf{T} A P$ is diagonal, $\mathrm{smry}_A(X) = \mathrm{smry}_{P^\mathsf{T} A P}(\tilde{X})$ holds for $\tilde{X} = [P^\mathsf{T} X^{ij} P]_{i,j=1,\ldots,n}$. □

4.2. (1) Implies (2)

Investigating whether K is positive semidefinite is equivalent to investigating whether the Gram matrices for finite subsets of χ are positive semidefinite. Therefore, without any loss of generality, we may assume that χ is a finite set $\{x_1, \ldots, x_n\}$. Since M_{x_i,x_j} are finite, we may also assume χ'_{x_i} are finite.

We slightly extend the definition of $(\chi', k, \{\gamma_x\})$ by adding a new element $\bullet \in \chi'$ such that $k(\bullet, \bullet) = k(\bullet, x') = k(x', \bullet) = 0$ hold for an arbitrary $x' \in \chi'$. Even after the extension, $(\chi', k, \{\gamma_x\})$ still remains an evaluating system for $\mathcal{M}$.

Next, we define $\bar{\chi}', \bar{M}$ and $\{\bar{\gamma}_x\}$ as follows: $\bar{\chi}'$ is the disjoint union $\bigsqcup_{i=1}^{n} \chi'_{x_i}$; $\bar{x}'$ denotes the image of $x' \in \chi'_x$ in $\bar{\chi}'$; $\bar{M} = \{(\bar{x}', \bar{y}') \mid (x', y') \in M_{x,y} \wedge x, y \in \chi\}$; $\bar{\gamma}_x : \bar{\chi}' \longrightarrow \chi'$ satisfies that $\bar{\gamma}_x(\bar{x}') = \gamma_x(x')$, if $x' \in \chi'_x$, and $\bar{\gamma}_x(\bar{x}') = \bullet$, otherwise. Then, the mapping kernel K with respect to $\mathcal{M}$ and $\mathcal{E}$ is rewritten as follows.

$$K(x, y) = \sum_{(\bar{x}', \bar{y}') \in \bar{M}} k(\bar{\gamma}_x(\bar{x}'), \bar{\gamma}_y(\bar{y}'))$$

Furthermore, $K(x_i, x_j) = \mathrm{tr}(A^\mathsf{T} X^{ij})$ holds, when we define m-dimensional matrices A and X^{ij} for $\bar{\chi}' =$

$\{\bar{x}'_1, \ldots, \bar{x}'_m\}$. $A_{kl} = 1$ if $(\bar{x}'_k, \bar{x}'_l) \in \bar{M}$, and $A_{kl} = 0$ otherwise; $X^{ij}_{kl} = k(\bar{\gamma}_{x_i}(\bar{x}'_k), \bar{\gamma}_{x_j}(\bar{x}'_l))$.

To show the assertion, it suffices to prove A is positive semidefinite by Proposition 1 ($X = [X^{ij}]_{i,j=1,\ldots,n}$ is positive semidefinite by definition). A is symmetric, since $(x', y') \in M_{x,y} \Leftrightarrow (y', x') \in M_{y,x}$ holds. The hypothesis that $\{M_{x,y}\}$ is transitive implies that $\{1, \ldots, m\}$ is decomposed into $U_1 \sqcup \cdots \sqcup U_M$ such that: $U_a \cap U_b = \emptyset$, if $a \neq b$; $(\bar{x}'_k, \bar{x}'_l) \in \bar{M}$, if, and only if, $k, l \in U_a$ for some $a \in \{1, \ldots, M\}$. Therefore, $A = \bigoplus_{a=1}^{M} A[U_a]$ holds, and therefore, A is positive semidefinite, since so are $A[U_a]$.

4.3. (2) Implies (1)

We prove the cotraposition of the assertion. If $\mathcal{M}$ is not transitive, A includes at least one of the following sub-matrices (without any loss of generality, we may assume $k < l < b$), where $A[i_1, \ldots, i_n]$ denote the n-dimensional matrix whose (α, β)-element is A_{i_α, i_β}.

$$A[k, l] = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{11}$$

$$A[k, l] = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \tag{12}$$

$$A[k, l, b] = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \tag{13}$$

Note that any of them has a negative eigenvalue, since $\det A < 0$ holds.

We will see that there exists an instance of $\mathcal{E} = (\chi', k, \{\gamma_x\})$ such that $\mathrm{smry}_A(X)$, which is the Gram matrix for χ, is not positive semidefinite, if any of the above three cases occurs. In the remaining of this section, we will give a proof only for the case where Eq. (13) holds. The assertion for the simpler cases, that is, where either Eq. (11) or (12) holds, can be proved in almost the same way.

Let i, j and a denote the indices such that $x'_k \in \chi'_{x_i}$, $x'_l \in \chi'_{x_j}$ and $x'_b \in \chi'_{x_a}$ (be reminded that $\bar{\chi}'$ is defined as the disjoint union of χ'_x for $x \in \chi$). The indices are not necessarily different from each other. Further, let column vectors $\vec{e}_1$, $\vec{e}_2$ and $\vec{e}_3$ be an orthogonal basis of $\mathbb{R}^3$ such that the following holds.

$$[\vec{e}_1, \vec{e}_2, \vec{e}_3]^\mathsf{T} A[k, l, b][\vec{e}_1, \vec{e}_2, \vec{e}_3] = \begin{bmatrix} \alpha_1 & 0 & 0 \\ 0 & \alpha_2 & 0 \\ 0 & 0 & \alpha_3 \end{bmatrix}$$

We assume $\alpha_1 < 0$ without any loss of generality, and define positive semidefinite K as follows.

$$K = [\vec{e}_1, \vec{e}_2, \vec{e}_3] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} [\vec{e}_1, \vec{e}_2, \vec{e}_3]^\mathsf{T}$$

$$\begin{aligned} &\therefore \ \mathrm{tr}(A[k, l, b]^\mathsf{T} K) \\ &= \ \mathrm{tr}\left(\begin{bmatrix} \alpha_1 & 0 & 0 \\ 0 & \alpha_2 & 0 \\ 0 & 0 & \alpha_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right) = \alpha_1 < 0 \end{aligned}$$

Now, we define $\mathcal{E} = (\chi', k, \{\gamma_x\})$ as follows.

- $\chi' = \{\bullet, \xi, \eta, \zeta\}$
- $\begin{bmatrix} k(\xi,\xi) & k(\xi,\eta) & k(\xi,\zeta) \\ k(\eta,\xi) & k(\eta,\eta) & k(\eta,\zeta) \\ k(\zeta,\xi) & k(\zeta,\eta) & k(\zeta,\zeta) \end{bmatrix} = K$
- $\gamma_x(x') = \begin{cases} \xi, & \text{if } x = x_i \text{ and } x' = x'_k, \\ \eta, & \text{if } x = x_j \text{ and } x' = x'_l, \\ \zeta, & \text{if } x = x_a \text{ and } x' = x'_b, \\ \bullet, & \text{otherwise.} \end{cases}$

Below, we investigate three cases: the indices take the same value, that is, $i = j = a$; two of the indices coincide with each other, where we can assume $i = j \neq a$ without loss of generality: the indices are different from one another, that is, $i \neq j \neq a \neq i$. For each case, we see that some diagonally located submatrix of $\mathrm{smry}_A(X)$ is not positive semidefinite. This implies that $\mathrm{smry}_A(X)$ itself is not positive semidefinite.

Case $i = j = a$: The submatrix $\mathrm{smry}_A(X)[i]$ is not positive semidefinite.

$$\mathrm{smry}_A(X)[i] = \mathrm{tr}(A[k, l, b]^\mathsf{T} K) < 0$$

Case $i = j \neq a$: We will show that $\mathrm{smry}_A(X)[i, k]$ is not positive semidefinite.

$$\begin{aligned} \mathrm{tr}(A^\mathsf{T} X^{ii}) &= \mathrm{tr}(A[k, l, b]^\mathsf{T}[1, 2] K[1, 2]) \\ \mathrm{tr}(A^\mathsf{T} X^{ia}) &= A[k, l, b]^\mathsf{T}{}_{1,3} K_{1,3} + A[k, l, b]^\mathsf{T}{}_{2,3} K_{2,3} \\ \mathrm{tr}(A^\mathsf{T} X^{ai}) &= A[k, l, b]^\mathsf{T}{}_{3,1} K_{3,1} + A[k, l, b]^\mathsf{T}{}_{3,2} K_{3,2} \\ \mathrm{tr}(A^\mathsf{T} X^{aa}) &= A[k, l, b]^\mathsf{T}{}_{3,3} K_{3,3} \end{aligned}$$

$$\therefore \mathrm{tr}\left(\mathrm{smry}_A(X)[i, a] \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \right) = \mathrm{tr}(A[k, l, b]^\mathsf{T} K) < 0$$

By Proposition 1 $\mathrm{smry}_A(X)[i, a]$ turns out not to be positive semidefinite.

Case $i \neq j \neq a \neq i$: For $\alpha, \beta = 1, 2, 3$, the (α, β)-element of $\mathrm{smry}_A(X)[i, j, a]$ coincides with

$A[k,l,b]^{\mathsf{T}}{}_{\alpha,\beta} K_{\alpha,\beta}$.

$$\begin{aligned} & \mathrm{tr}\left(\mathrm{smry}_A(X)[i,j,a]\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}\right) \\ = \quad & \mathrm{tr}(A[k,l,b]^{\mathsf{T}} K) < 0 \end{aligned}$$

By Proposition 1, $\mathrm{smry}_A(X)[i,j,a]$ turns out not to be positive semidefinite.

References

Aoki, K. F., Yamaguchi, A., Okuno, Y., Akutsu, T., Ueda, N., Kanehisa, M., & Mamitsuka, H. (2003). Efficient tree-matching methods for accurate carbohydrate database query. *Genome Informatics*, *14*, 134 – 143.

Barnard, D., Clarke, G., & Duncan, N. (1995). *Tree-to-tree correction for document trees* (Technical Report 95-375). Queen's University, Kingston, Ontario K7L 3N6 Canada.

Berry, V., & Nicolas, F. (2004). Maximum Agreement and Compatible Supertrees (Extended Abstract). *CPM* (pp. 205–219).

Bunke, H. (1997). On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, *18*, 689–694.

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

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press.

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

Hein, J., Jiang, T., Wang, L., & Zhang, K. (1996). On the complexity of comparing evolutionary trees. *Discrete Applied Mathematics*, *71*, 153 – 169.

Jansson, J., Ng, J. H. K., Sadakane, K., & Sung, W. K. (2005). Rooted maximum agreement supertrees. *Algorithmica*, 293 – 307.

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

Kuboyama, T., Shin, K., & Kashima, H. (2006). Flexible tree kernels based on counting the number of tree mappings. *Proc. of Machine Learning with Graphs*.

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

Li, H., & Jiang, T. (2005). A class of edit kernels for svms to predict translation initiation sites in eukaryotic mrnas. *Trans. on Comput. Syst. Bio. II*, *LNBI 3680*, 48 – 58.

Lodhi, H., Shawe-Taylor, J., Cristianini, N., & Watkins, C. J. C. H. (2001). Text classificatio using string kernels. *Advances in Neural Information Processing Systems*, *13*.

Lu, C. L., Su, Z.-Y., & Tang, G. Y. (2001). A New Measure of Edit Distance between Labeled Trees. *LNCS* (pp. pp. 338–348). Springer-Verlag Heidelberg.

Menchetti, S., Costa, F., & Frasconi, P. (2005). Weighted decomposition kernel. *Proc. of the 22nd International Conference on Machine Learning*.

Pelillo, M. (2002). Matching free trees, maximal cliques, and monotone game dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*, 1535 – 1541.

Salzberg, S. L. (1997). A method for identifying splice sites and translational staqr sites in eukaryotic mrna. *Computer Applications in the Biosciences*, *13*, 365 – 376.

Suzuki, J., Isozaki, H., & Maeda, E. (2004). Convolution kernels with feature selection for natural language processing tasks. *Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 119–126).

Tai, K. C. (1979). The Tree-to-Tree Correction Problem. *JACM*, *26*, 422–433.

Wagner, R., & Fischer, M. (1974). The string-to-string correction problem. *JACM*, *21*, 168–173.

Zhang, K. (1995). Algorithms for the constrained editing distance between ordered labeled trees and related problems. *PR*, *28*, 463–474.

Zien, A., Rätsch, G., Mika, S., Schölkopf, B., Lengauer, T., & Müller, K. R. (2000). Engineering support vector machne kernels that recognize translation initiation sites. *Bioinformatics*, *16*, 799 – 807.