To appear in *Advanced Robotics*
Vol. 00, No. 00, January 2013, 1–24

FULL PAPER

Efficient Learning Algorithm for Sparse SubSequence Pattern-based Classification and Applications to Comparative Animal Trajectory Data Analysis

Takuto Sakuma[a], Kazuya Nishi[a], Kaoru Kishimoto[a], Kazuya Nakagawa[a], Masayuki Karasuyama[abc], Yuta Umezu[a], Shinsuke Kajioka[a], Shuhei J. Yamazaki[d], Koutarou D. Kimura[de], Sakiko Matsumoto[f], Ken Yoda[f], Matasaburo Fukutomi[g], Hisashi Shidara[h], Hiroto Ogawa[h] and Ichiro Takeuchi[ibj*]

[a] *Department of Computer Science, Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya, 466-8555, Japan*;
[b] *Center for Materials Research by Information Integration, National Institute for Materials Science, 1-2-1 Sengen, Tsukuba, 305-0047, Japan*;
[c] *JST, PRESTO, 4-1-8 Honcho, Kawaguchi, Saitama, 332-0012, Japan* [d] *Department of Biological Sciences, Graduate School of Science, Osaka University, 1-1 Machikane-yama, Toyonaka, Osaka, 560-0043, Japan*;
[e] *Graduate School of Natural Sciences, Nagoya City University, 1 Yamanohata, Mizuho-cho, Mizuho-ku, Nagoya, 467-8501. Japan*;
[f] *Graduate School of Environmental Studies, Nagoya University, Furo-cho, Chikusa-ku, Nagoya, 464-8601. Japan*;
[g] *Graduate School of Life Science, Hokkaido University, Kita 10, Nishi 8, Kita-ku, Sapporo, 060-0810, Japan*;
[h] *Department of Biological Sciences, Faculty of Science, Hokkaido University, Kita 10, Nishi 8, Kita-ku, Sapporo, 060-0810, Japan*;
[i] *Department of Computer Science/Research Institute for Information Science, Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya, 466-8555, Japan*;
[j] *RIKEN Center for Advanced Intelligence Project, 1-4-1 Nihonbashi, Chuo-ku, Tokyo, 103-0027, Japan*;
(*Received 00 Month 201X; accepted 00 Month 201X*)

Recent advances in robotics and measurement technologies have enabled biologists to record the trajectories created by animal movements. In this paper, we convert time series of animal trajectories into sequences of finite symbols, and then propose a machine learning method for gaining biological insight from the trajectory data in the form of symbol sequences. The proposed method is used for training a classifier which differentiates between the trajectories of two groups of animals such as male and female. The classifier is represented in the form of a sparse linear combination of subsequence patterns, and we call the classifier an *S3P-classifier*. The trained S3P-classifier is easy to interpret because each coefficient represents the specificity of the subsequence patterns in either of the two classes of animal trajectories. However, fitting an S3P-classifier is computationally challenging because the number of subsequence patterns is extremely large. The main technical contribution in this paper is the development of a novel algorithm for overcoming this computational difficulty by combining a sequential mining technique with a recently developed convex optimization technique called *safe screening*. We demonstrate the effectiveness of the proposed method by applying it to three animal trajectory data analysis tasks.

Keywords: animal behaviors, animal trajectories, discriminative sequential pattern mining

*Corresponding author. Email: takeuchi.ichiro@nitech.ac.jp

1. Introduction

Recent advances in robotics and measurement technologies have enabled biologists to record comprehensive and exhaustive data on animal trajectories. For example, by using small logging devices attached to animals, the trajectories of wild animal movements can be recorded in detail. This approach is called *bio-logging*, and is considered a promising framework for wildlife animal behavior research. A similar approach is used in laboratory studies of micro-scale model organisms. For example, by using a high-speed auto-tracking microscopic system, the detailed trajectories of micro organisms in designed experiments can be recorded. These animal trajectory big data can be used for expanding scientific knowledge in the field of biology in a data-driven manner by using machine learning and artificial intelligence-based data analysis techniques. In this paper, we study machine learning methods for analyzing animal trajectory big data.

It is important in biological studies to identify behaviors which differ between two groups of animals. For example, in bio-logging studies, researchers are interested in finding differences in the behaviors of female and male animals, or adult and child animals. Similarly, molecular biologists are interested in identifying different behaviors in wild-type and mutant strains for investigating the functions of certain genetic factors. Animal trajectory data are typically represented as time series. In this paper, we study machine learning methods for classifying two groups of animal trajectories. Since biologists are interested not only in classification but also in interpretation, the classifier should have a high interpretability as well as a high classification accuracy.

In this paper, we propose a new animal trajectory classification model called *sparse subsequence pattern-based classifiers (S3P-classifiers)*, which is based on a discriminant function in the form of sparse linear combinations of multiple subsequence patterns. Figure 1 shows a schematic illustrating how we learn the proposed S3P-classifiers (the formal problem setup is described in the next section). We consider a situation where two time series are available for the training data (see (a)). Each time series is represented as a sequence of a finite number of symbols (see (b)). From the two groups of symbol sequences, we learn a discriminant function in the form of sparse linear combinations of the existences of subsequence patterns (see (c)). The trained discriminant function is then used for classifying a new symbol sequence (see (d)). Note that although the number of all possible subsequence patterns is extremely large, the discriminant function in (c) only has a small number of terms under the sparsity requirement. The subsequence patterns having positive/negative coefficients can be interpreted as appearing more frequently in the first/second group of animals respectively (see (e)).

Fitting an optimal sparse subsequence pattern-based model as in Fig. 1 (c) is computationally challenging because the number of possible subsequence patterns is extremely large. For example, when the number of different symbols (e.g., F, B, L, R) is 4, the length of the entire sequence is 500, and if subsequence patterns of up to length 50 are considered, the number of all possible subsequence patterns is roughly 1.3e+30. Our main technical contribution in this paper is to develop a novel algorithm for resolving this computational difficulty. To this end, we effectively combine a safe screening method developed in the convex optimization community [1–9] and a sequential mining method developed in the data mining community [10–21]. Safe screening methods can be used for finding a subset of features which cannot be active in the optimal sparse model before actually learning the optimal model. In order to exploit the advantages of safe screening methods for our sparse subsequence pattern-based modeling, we extend one of the safe screening methods so that it can be used together with a subsequence search algorithm defined in a search tree. This extension allows us to efficiently screen out a large number of subsequence patterns that are guaranteed to be irrelevant in the optimal model. This then enables us to learn the optimal S3P-classifier without handling all possible subsequence patterns. We note that Nakagawa et al. [22] recently combined a safe screening method with the itemset mining method to efficiently identify sparse high-order interaction models. Our main contribution in this paper is to adapt the techniques in Nakagawa et al. to the problem of learning sparse subsequence

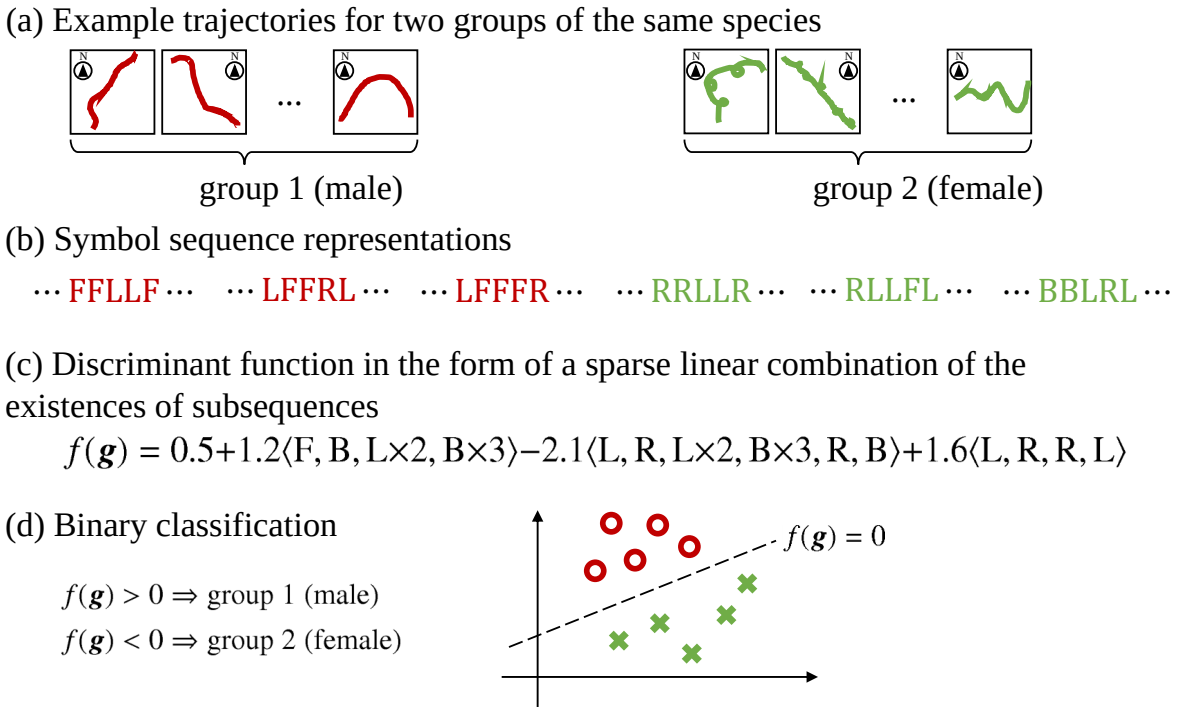

Figure 1. Schematic example of the proposed classifier building procedure. (a) We consider a situation where two groups of animal behavior records (e.g., male vs. female) are available as a training data set. (b) Each animal behavior record is represented as a sequence of symbols (e.g., F, B, R, and L, each of which represents a **F**orward, **B**ackward, **R**ight, and **L**eft movement). (c) We build a discriminant function $f(\boldsymbol{g})$ for classifying a symbol sequence $\boldsymbol{g}$ to the 1st class (male) or the 2nd class (female). The discriminant function has the form of a sparse linear combination of the existences of subsequence patterns. Note that, by the sparsity requirement, only the coefficients of subsequence patterns $\langle \mathrm{F}, \mathrm{B}, \mathrm{L}\times 2, \mathrm{B}\times 3\rangle, \langle \mathrm{L}, \mathrm{R}, \mathrm{L}\times 2, \mathrm{B}\times 3, \mathrm{R}, \mathrm{B}\rangle, \langle \mathrm{L}, \mathrm{R}, \mathrm{R}, \mathrm{L}\rangle$ have non-zero values. (d) The trained discriminant function $f(\boldsymbol{g})$ is used for classifying animal behavior $\boldsymbol{g}$ in the form of a symbol sequence by the sign of the discriminant function value. (e) The signs of the coefficients in the discriminant function indicate that the subsequence patterns $\langle \mathrm{F}, \mathrm{B}, \mathrm{L}\times 2, \mathrm{B}\times 3\rangle$ and $\langle \mathrm{L}, \mathrm{R}, \mathrm{R}, \mathrm{L}\rangle$ appear more frequently in male animals, while the subsequence pattern $\langle \mathrm{L}, \mathrm{R}, \mathrm{L}\times 2, \mathrm{B}\times 3, \mathrm{R}, \mathrm{B}\rangle$ appears more frequently in female animals.

classifiers and apply it to animal trajectory data analysis.

The rest of the paper is organized as follows. Section 2 presents the problem setup and overviews the sequential pattern mining method. Section 3 describes our main contribution where we introduce a new algorithm for fitting S3P-classifiers. In Section 4, we apply the S3P-classifiers to published animal trajectory datasets for three animals, for streaked shearwater, the nematode *C. elegans*, and crickets. Section 5 concludes the paper. A preliminary version of this paper will be presented at a conference [23], in which we merely show the results when using conventional frequent sequential mining algorithms for extracting frequent animal movement behaviors in each group of animals. The main differences between this submission and [23] are that in this paper we develop a classifier for explaining the differences in the movement behaviors between two groups of animals, and introduce a novel learning algorithm for obtaining the classifier.

Notation

We use the following notation in the rest of the paper. For any natural number n, we define $[n] := \{1, \ldots, n\}$. For an n-dimensional vector $\boldsymbol{v}$ and a set $\mathcal{I} \subseteq [n]$, $\boldsymbol{v}_{\mathcal{I}}$ represents a subvector of $\boldsymbol{v}$ whose elements are indexed by $\mathcal{I}$. The indicator function is written as $I(\cdot)$; i.e., $I(z) = 1$ if z is true, and $I(z) = 0$ otherwise. The L1 norm of a vector $\boldsymbol{v}$ is written as $\|\boldsymbol{v}\|_1$. A sequence (an ordered list of discrete symbols) with length T is represented as $\langle g_1, g_2, \ldots, g_T\rangle$.

Table 1. Illustrative example dataset with $n = 6(n_+ = n_- = 3)$, $T(1) = 4$, $T(2) = 4$, $T(3) = 7$, $T(4) = 4$, $T(5) = 4$, $T(6) = 7$, and $\mathcal{S} = \{a, b, c, d, e, f\}$. Based on this type of training set, we learn a S3P-classifier whose discriminant function is represented as a sparse linear combination of the patterns.

Sequence	Label
$\boldsymbol{g}_1 = \langle a, d, e, a \rangle$	male
$\boldsymbol{g}_2 = \langle e, a, d, f \rangle$	male
$\boldsymbol{g}_3 = \langle e, a, b, f, b, d, c \rangle$	male
$\boldsymbol{g}_4 = \langle a, a, b, f \rangle$	female
$\boldsymbol{g}_5 = \langle e, b, b, b \rangle$	female
$\boldsymbol{g}_6 = \langle a, b, f, b, b, c, e \rangle$	female

2. Preliminaries

We first formulate our problem setting.

2.1 *Problem Setup*

In general, an individual animal trajectory is recorded as a time series. We assume that appropriate preprocessing operations such as outlier removal, missing value imputation, and noise reduction have been applied to the raw data before analyzing the time series. For developing a S3P-classifier, a time series is first transformed into a sequence by discretization [24–27] [1]. A sequence is an ordered list of discrete symbols. We denote the number of different symbols as m and denote the set of those symbols as $\mathcal{S} := \{s_1, \ldots, s_m\}$. In this paper, we consider two groups of animals such as male/female or infant/adults. Let the total number of animals be n.We denote the first and the second group of animals as $\mathcal{G}_+, \mathcal{G}_- \subseteq [n]$ and their sizes as $n_+ := |\mathcal{G}_+|, n_- := |\mathcal{G}_-|$, respectively. The training set for learning an S3P-classifier is written as

$$\{(\boldsymbol{g}_i, y_i)\}_{i \in [n]},$$

where $\boldsymbol{g}_i$ represents the sequence of the i-th animal, and $y_i \in \{\pm 1\}$ represents the label of each group, i.e., $y_i = +1$ if the i-th animal is in the first group, and $y_i = -1$ if the i-th animal is in the second group. Each sequence $\boldsymbol{g}_i$ is written as

$$\boldsymbol{g}_i := \langle g_{i1}, g_{i2}, \ldots, g_{iT(i)} \rangle, i \in [n],$$

where g_{it} represents the symbol of the i-th animal at the t-th time point which takes one of the symbols in $\mathcal{S}$, and $T(i)$ indicates the length of the i-th sequence. Table 1 shows an example dataset.

We call a segment of a movement behavior record a *pattern*. We denote patterns as $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots$, each of which is also defined as a symbol sequence of the form

$$\boldsymbol{q}_j := \langle q_{j1}, q_{j2}, \ldots, q_{jL(j)} \rangle, j = 1, 2, \ldots,$$

where $L(j)$ is the length of the pattern $\boldsymbol{q}_j$ for $j = 1, 2, \ldots$. We say that a sequence $\boldsymbol{g}_i$ contains a pattern $\boldsymbol{q}_j$ if

$$\exists \{1 \leq i_1 < \ldots < i_{L(j)} \leq T(i)\} \text{ such that } q_{j1} = g_{i_1}, q_{j2} = g_{i_2}, \ldots, q_{jL(j)} = g_{i_{L(j)}},$$

[1] A time series is an ordered list of numbers, whereas a sequence is an ordered list of nominal values (symbols).

and represent this relationship as $\boldsymbol{q}_j \sqsubseteq \boldsymbol{g}_i$. We denote the set of all possible patterns contained in any one of the sequences $\{\boldsymbol{g}_i\}_{i\in[n]}$ as $\mathcal{Q} := \{\boldsymbol{q}_i\}_{i\in[d]}$, where d is the number of possible patterns. Note that the size of $\mathcal{Q}$ is quite large in general.

In this paper, we introduce a classifier based on sparse linear combinations of patterns

$$f(\boldsymbol{g}_i; \mathcal{Q}) := \sum_{\boldsymbol{q}_j \in \mathcal{Q}} w_j I(\boldsymbol{q}_j \sqsubseteq \boldsymbol{g}_i) + b, \tag{1}$$

where $w_j \in \mathbb{R}$ and $b \in \mathbb{R}$ are the parameters of the linear model. We estimate these parameters by solving the following minimization problem

$$\min_{\boldsymbol{w},b} \sum_{i\in[n]} \ell(y_i, f(\boldsymbol{g}_i; \mathcal{Q})) + \lambda \|\boldsymbol{w}\|_1, \tag{2}$$

where $\boldsymbol{w} := [w_1, \ldots, w_d]^\top$, $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a loss function, and $\lambda > 0$ is a tuning parameter. In (2), by introducing the L_1 penalty term $\|\boldsymbol{w}\|_1$, the solution of the minimization problem is induced to be sparse; i.e., many coefficients w_js are zero in the optimal solution. This approach is called sparse learning. Statistical properties and optimization algorithms of sparse learning have been intensively studied in the fields of machine learning and statistics [28]. Sparse learning is useful for building an interpretable model because it allows us to conduct model building and feature (pattern) selection simultaneously. We note that existing optimization algorithms for sparse learning cannot be used for solving the minimization problem in (2) since the number of patterns $|\mathcal{Q}|$ is quite large. In the next section, we propose a novel optimization algorithm by combining the safe screening and sequential mining techniques.

2.2 *Sequential Pattern Mining*

Sequential pattern mining methods are widely used for extracting frequently occurring subsequences from a set of sequences. In the next section we use a frequent sequential mining technique as a building block for our learning algorithm. In this section, we first describe the basic idea of sequential pattern mining, and then present a famous sequential pattern mining algorithm called PrefixSpan. In this paper, we focus on finding contiguous patterns, i.e., patterns having no breaks within the pattern. It is easy to extend this method to the case of discontiguous sequential patterns.

2.2.1 Frequent sequential pattern mining

The database of sequences is denoted by $\mathcal{D} := \{\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_n\}$. We define the support of the pattern $\boldsymbol{q}_j$ as

$$\text{support}(\boldsymbol{q}_j) := |\{\boldsymbol{g}_i \mid \boldsymbol{g}_i \in \mathcal{D} \text{ and } \boldsymbol{q}_j \sqsubseteq \boldsymbol{g}_i\}|,$$

where $\text{support}(\boldsymbol{q}_j)$ indicates the number of sequences that contain the pattern $\boldsymbol{q}_j$. The set of all patterns that appears min_sup or more times is called *frequent sequential patterns* and denoted as

$$F(\text{min\_sup}) := \{\boldsymbol{q}_j \in \mathcal{Q} \mid \text{support}(\boldsymbol{q}_j) \geq \text{min\_sup}\}.$$

In the context of pattern mining, the threshold value min_sup is called the *minimum support*. A method that can find frequent sequential patterns is called a *frequent sequential pattern mining* method. For example, in Table 1, when min_sup = 2,

$$F(3) = \{\langle a\rangle, \langle b\rangle, \langle d\rangle, \langle e\rangle, \langle f\rangle, \langle a, b\rangle, \langle b, f\rangle, \langle e, a\rangle, \langle a, b, f\rangle\}.$$

Since the number of possible patterns $|\mathcal{Q}|$ is quite large in general, it is often infeasible to actually count the supports of all possible patterns. To circumvent this difficulty, sequential pattern mining methods exploit the fact that the support of a pattern is always less than or equal to the supports of any of its subsequences. Consider two sequences $\boldsymbol{q}_{j'}$ and $\boldsymbol{q}_j$ such that $\boldsymbol{q}_{j'} \sqsubseteq \boldsymbol{q}_j$, i.e., $\boldsymbol{q}_{j'}$ is a subsequence of $\boldsymbol{q}_j$; then, it is obvious that

$$\text{support}(\boldsymbol{q}_{j'}) \geq \text{support}(\boldsymbol{q}_j) \quad \forall \boldsymbol{q}_{j'} \sqsubseteq \boldsymbol{q}_j. \tag{3}$$

Equation (3) indicates that, when we consider a tree as in Fig. 2, the support of the pattern in a node is always greater than or equal to its descendant node patterns, and less than or equal to its ancestor node patterns. This anti-monotonicity of the support in the tree can be exploited to find frequent sequential patterns. Namely, when we search over the tree, if the support of a node in the tree is already smaller than min_sup, we can skip searching over its subtree.

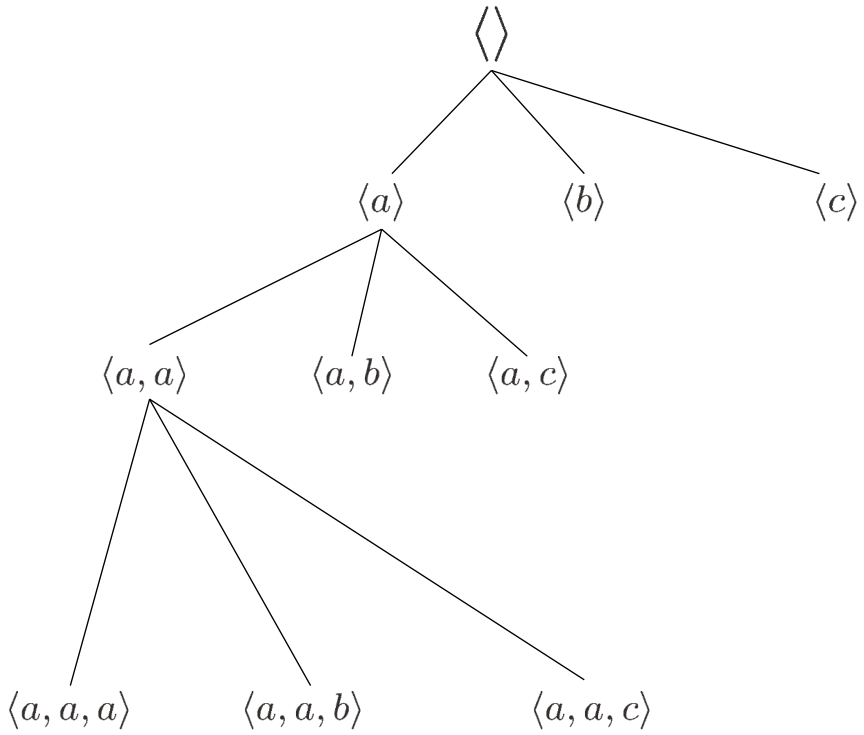

Figure 2. Tree structure for a frequent sequential pattern mining problem with $\mathcal{S} = \{a, b, c\}$.

2.2.2 PrefixSpan

In the data mining literature, several types of sequential pattern mining methods have been proposed [20]. From these methods, we use a pattern-growth type method, and employ the most popular algorithm called PrefixSpan [10]. The PrefixSpan algorithm begins by exploring the search space of sequential patterns based on a depth-first search. Then, starting from the sequential patterns containing only a single symbol, it explores longer patterns by recursively appending symbols to the existing patterns.

To formulate the PrefixSpan algorithm, we first define a concatenation of a sequence and a symbol. Given a sequence $\boldsymbol{z} = \langle z_1, z_2, \ldots, z_T \rangle$ and a symbol $s \in \mathcal{S}$, the notation

$$\boldsymbol{z} \diamond s = \langle z_1, z_2, \ldots, z_T, s \rangle$$

indicates the concatenation of $\boldsymbol{z}$ and s. Given two sequences $\boldsymbol{h} = \langle h_1, h_2, \ldots, h_{T_1} \rangle$ and $\boldsymbol{o} =$

$\langle o_1, o_2, \ldots, o_{T_2} \rangle$, the concatenation of $\boldsymbol{h}$ and $\boldsymbol{o}$ is then

$$\boldsymbol{h} \diamond \boldsymbol{o} = \langle h_1, h_2, \ldots, h_{T_1}, o_1, o_2, \ldots, o_{T_2} \rangle.$$

In the PrefixSpan algorithm, the reduced database obtained by removing a specific sequence $\boldsymbol{z}$ as a prefix from the original database $\mathcal{D}$ is defined as the *projected database* $\mathcal{D}_{\boldsymbol{z}}$ and defined as

$$\mathcal{D}_{\boldsymbol{z}} := \{\boldsymbol{o} | \boldsymbol{z}' \in \mathcal{D}, \boldsymbol{z}' = \boldsymbol{h} \diamond \boldsymbol{o}\} \quad \text{s.t. } \boldsymbol{z} \sqsubseteq \boldsymbol{h} \text{ and } \nexists \boldsymbol{h}', \boldsymbol{z} \sqsubseteq \boldsymbol{h}' \sqsubset \boldsymbol{h}, \tag{4}$$

where $\boldsymbol{h}$ represents the smallest prefix including $\boldsymbol{z}$ in $\boldsymbol{z}'$.

Typically, in sequential pattern mining problems, the contiguity of patterns is not considered, and only the order of the symbols matters. For example, in general sequential pattern mining contexts, both of the sequences $\boldsymbol{z}_1 = \langle s_1, s_2, s_3 \rangle$ and $\boldsymbol{z}_2 = \langle s_1, s_4, s_5, \ldots, s_{100}, s_2, s_3 \rangle$ are considered to contain a sequential pattern $\boldsymbol{q} = \langle s_1, s_2, s_3 \rangle$. In this paper, however, we focus on finding contiguous patterns, and regard that $\boldsymbol{z}_2$ does not contain $\boldsymbol{q}$ in the above example. To reflect this change, we need to slightly change the definition of the projected database to

$$\mathcal{D}_{\boldsymbol{z}} = \{\boldsymbol{o} | \boldsymbol{z}' \in \mathcal{D}, \boldsymbol{z}' = \boldsymbol{h} \diamond \boldsymbol{o}\} \quad \text{s.t. } \boldsymbol{z} \sqsubseteq \boldsymbol{h}, |\boldsymbol{z}| = |\boldsymbol{h}|, \text{ and } \nexists \boldsymbol{h}', \boldsymbol{z} \sqsubseteq \boldsymbol{h}' \sqsubset \boldsymbol{h}. \tag{5}$$

In the example in Table 1, the projected databases in our definitions are given as

$$\mathcal{D}_{\langle e,a \rangle} = \{\langle d, f \rangle, \langle b, f, b, d, c \rangle\},$$
$$\mathcal{D}_{\langle a,b \rangle} = \{\langle f, b, d, c \rangle, \langle f \rangle, \langle f, b, b, c, e \rangle\}.$$

The pseudo-code of the PrefixSpan algorithm is presented in Algorithm 1. The PrefixSpan algorithm is efficient since only the sequential patterns appearing more than min_sup times in $\mathcal{D}$ are selected and counted.

Algorithm 1 PrefixSpan($\boldsymbol{z}$, $\mathcal{D}_{\boldsymbol{z}}$, min_sup, F(min_sup))

Require: A sequence $\boldsymbol{z}$, and a projected Database $\mathcal{D}_{\boldsymbol{z}}$, min_sup.
Ensure: The frequent sequence set F(min_sup).

```
insert z to F(min_sup)
scan D_z once, find every frequent symbol h such that z can be extended to (z ◇ h);
if There is no valid h then
    return;
end if
for h ∈ H do
    Call PrefixSpan(z ◇ h, D_{z◇h}, min_sup, F(min_sup));
end for
```

3. Learning Sparse SubSequence Pattern-based Classifier

In this section, we introduce a method for efficiently solving the optimization problem in (2) by effectively combining the sequential pattern mining technique introduced in the previous section with a convex optimization technique called safe screening. Safe screening is a method for pre-screening a subset of variables to see whether they are required for the optimal model before solving the optimization problem. It allows one to remove those variables from the optimization in advance, and hence the reduced optimization problem can be solved efficiently. In Section 3.1, we demonstrate how the safe screening technique can be used for learning a sparse classifier. Then, in Section 3.2, we develop a novel algorithm for exploiting the safe screening technique in our problem setup, where we effectively use the tree structure among patterns and the search

strategy in sequential mining. We note that Nakagawa et al. [22] recently combined the safe screening method with the itemset mining method for efficiently identifying sparse high-order interaction models. Our main contribution in this paper is to adapt the techniques in [22] to the problem of learning the S3P-classifier.

3.1 *Safe Screening for Discriminative Models*

In safe screening, unnecessary variables can be detected beforehand and reduced before solving the optimization problem. This corresponds to finding j such that $w_j = 0$ in the optimal solution $\boldsymbol{w}^* := [w_1^*, \ldots, w_{|\mathcal{Q}|}^*]^\top$ in the optimization problem (2). Such w_j does not affect the optimal solution even if it is removed beforehand.

Remark 1. *In the optimal solution $\boldsymbol{w}^*$ of the optimization problem (2), a set of j such that $|w_j^*| > 0$ is called the active set, and denoted as $\mathcal{A} \subseteq [|\mathcal{Q}|]$. In this case, even if only the subsequence patterns included in $\mathcal{A}$ are used, the same optimal solution as when using all the subsequence patterns can be obtained. Thus, if one solves*

$$(\boldsymbol{w}'^*_{\mathcal{A}}, b'^*) := \operatorname*{argmin}_{\boldsymbol{w},b} \sum_{i\in[n]} \ell(y_i, f(\boldsymbol{g}_i; \{\boldsymbol{q}\}_{i\in\mathcal{A}})) + \lambda\|\boldsymbol{w}\|_1, \tag{6}$$

then it is guaranteed that $\boldsymbol{w}^*_{\mathcal{A}} = \boldsymbol{w}'^*_{\mathcal{A}}$ and $b^* = b'^*$.

Therefore, if one has a knowledge that $w_j^* = 0$, by removing w_j from the optimization problem beforehand, it is possible to reduce the optimization problem and improve its efficiency without losing the optimality. In this section, we describe a method for finding such w_j before solving the optimization problem.

Let us define the feature vector $\boldsymbol{x}_i := [x_{i1}, x_{i2}, \ldots, x_{id}]$ for the ith sequence $\boldsymbol{g}_i$ as

$$x_{ij} := I(\boldsymbol{q}_j \sqsubseteq \boldsymbol{g}_i), \qquad\qquad j = 1, \ldots, |\mathcal{Q}|.$$

If the loss function is defined as $\ell(y, f) = (y - f)^2$, the optimization problem (2) is reduced to the Lasso problem [29] as follows:

$$\min_{\boldsymbol{w},b} \sum_{i\in[n]} \Big(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i + b\Big)^2 + \lambda\|\boldsymbol{w}\|_1. \tag{7}$$

Alternatively, one can use the so-called squared hinge-loss function $\ell(y, f) = \max\{0, 1 - yf\}^2$

$$\min_{\boldsymbol{w},b} \sum_{i\in[n]} \max\Big\{0, 1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)\Big\}^2 + \lambda\|\boldsymbol{w}\|_1, \tag{8}$$

for fitting a binary classifier. For the optimization problems (7) and (8), the dual problem can be written as

$$\begin{aligned} &\max_{\boldsymbol{\theta}} -\frac{\lambda^2}{2}\|\boldsymbol{\theta}\|_2^2 + \lambda\boldsymbol{\delta}^\top\boldsymbol{\theta} \\ &\text{s.t.} \left|\sum_{i\in[n]} \alpha_{ij}\theta_i\right| \le 1,\ j \in [d] \\ &\qquad \boldsymbol{\beta}^\top\boldsymbol{\theta} = 0, \end{aligned} \tag{9}$$

where $\alpha_{ij} = x_{ij}, \beta_i = 1$, $i \in [n]$, $j \in [d]$ for the problem in (7), and $\alpha_{ij} = y_i x_{ij}, \beta_i = y_i$, $i \in [n]$, $j \in [d]$ for the problem in (8).

If the primal and the dual optimal solutions are $\boldsymbol{w}^*$ and $\boldsymbol{\theta}^* := [\theta_1^*, \ldots, \theta_d^*]^\top$, respectively, the following theorem holds from the Karush-Kuhn-Tucker (KKT) optimality conditions.

Lemma 1. *For the optimal parameter w_j^* of an arbitrary subsequence $\boldsymbol{q}_j$, the following relationship holds:*

$$\left| \sum_{i \in [n]} \alpha_{ij} \theta_i^* \right| < 1 \Rightarrow w_j^* = 0.$$

The proof of Lemma 1 is presented in Appendix A.1. If the left-hand side of the above inequality is less than 1, then we know without solving the optimization problem and calculating $\boldsymbol{w}^*$ and $\boldsymbol{\theta}^*$ that the corresponding w_j is 0. In order to derive an upper bound for the left-hand side, a range where the optimal $\boldsymbol{\theta}^*$ exists can be derived based on information on a primal feasible solution $(\tilde{\boldsymbol{w}}, \tilde{b})$ and a dual feasible solution $\tilde{\boldsymbol{\theta}}$. Defining the objective function value of the primal problem with $(\tilde{\boldsymbol{w}}, \tilde{b})$ as $P_\lambda(\tilde{\boldsymbol{w}}, \tilde{b})$ and the objective function value of the dual problem with $\tilde{\boldsymbol{\theta}}$ as $D_\lambda(\tilde{\boldsymbol{\theta}})$, the following lemma holds:

Lemma 2 (Theorem 3 in [9])**.** *Let $(\tilde{\boldsymbol{w}}, \tilde{b})$ be an arbitrary primal feasible solution, and $\tilde{\boldsymbol{\theta}}$ be an arbitrary dual feasible solution. Then, the dual optimal solution $\boldsymbol{\theta}^*$ is within the ball in the dual solution space $\mathbb{R}^n$ with center $\tilde{\boldsymbol{\theta}}$ and radius $r_\lambda := \sqrt{2(P_\lambda(\tilde{\boldsymbol{w}}, \tilde{b}) - D_\lambda(\tilde{\boldsymbol{\theta}}))/\lambda}$.*

See Theorem 3 and its proof in [9]. This lemma tells us that, given a pair of primal feasible and dual feasible solutions, we can bound the dual optimal solution within a ball.

Lemma 2 can be used to derive an upper bound for $|\sum_{i \in [n]} \alpha_{ij} \theta_i^*|$. Since we know that the dual optimal solution $\boldsymbol{\theta}^*$ is within the ball in Lemma 2, an upper bound of any j for $\boldsymbol{q}_j \in \mathcal{Q}$ can be obtained by solving the following convex optimization problem:

$$\begin{aligned} \mathrm{UB}(j) := \max_{\boldsymbol{\theta} \in \mathbb{R}^n} & \left| \sum_{i \in [n]} \alpha_{ij} \theta_i \right| \\ \text{s.t.} & \left\| \boldsymbol{\theta} - \tilde{\boldsymbol{\theta}} \right\|_2 \leq \sqrt{2(P_\lambda(\tilde{\boldsymbol{w}}, \tilde{b}) - D_\lambda(\tilde{\boldsymbol{\theta}}))/\lambda}, \\ & \boldsymbol{\beta}^\top \boldsymbol{\theta} = 0. \end{aligned} \tag{10}$$

Fortunately, the convex optimization problem (10) can be explicitly solved as in the following lemma.

Lemma 3. *The solution of the convex optimization problem (10) is given as*

$$\mathrm{UB}(j) = \left| \sum_{i \in [n]} \alpha_{ij} \tilde{\theta}_i \right| + r_\lambda \sqrt{\sum_{i \in [n]} \alpha_{ij}^2 - \frac{(\sum_{i \in [n]} \alpha_{ij} \beta_i)^2}{\|\boldsymbol{\beta}\|_2^2}}.$$

The proof of Lemma 3 is presented in Appendix A.2.

3.2 *Safe Subsequence Pruning*

By evaluating Lemma 3 based on some feasible solution, it is possible to eliminate the unnecessary weight parameter w_j without directly solving the optimization problem. However, it is difficult to calculate the upper bound in Lemma 3 for a huge number of possible subsequence

patterns. Therefore, we propose a rule for removing multiple unnecessary subsequence patterns simultaneously by pruning the search tree for sequential pattern mining. We call this rule the safe subsequence pruning (SSP) rule. Let $\mathcal{C} := [d]$ be the index for the entire set of nodes included in the search tree (Fig. 2) of the sequential pattern mining. Also, let $\mathcal{C}_{\mathrm{sub}}(c) \subset \mathcal{C}$ be a subtree of $\mathcal{C}$ with root $c \in \mathcal{C}$. The SSP rule can be evaluated at each node in the tree, and if the condition is satisfied, it can be guaranteed that none of the subsequence patterns included in the subtree $\mathcal{C}_{\mathrm{sub}}(c)$ can be active in the optimal solution. This strategy allows us to delete a large number of subsequence patterns simultaneously.

The SSP rule can be derived as in the following theorem;

Theorem 1. *Given an arbitrary primal feasible solution* $(\tilde{\boldsymbol{w}}, \tilde{b})$ *and an arbitrary dual feasible solution* $\tilde{\boldsymbol{\theta}}$*, for any node* $c' \in \mathcal{C}_{\mathrm{sub}}(c)$*, the following Safe Subsequence Pruning Criterion* (SSPC) *provides the following rule*

$$\mathrm{SSPC}(c) := u_c + r_\lambda \sqrt{v_c} < 1 \Rightarrow w^*_{c'} = 0, \tag{11}$$

where

$$u_c := \max\left\{ \sum_{i:\beta_i \tilde{\theta}_i > 0} \alpha_{ic} \tilde{\theta}_i, \sum_{i:\beta_i \tilde{\theta}_i < 0} \alpha_{ic} \tilde{\theta}_i \right\}, v_c := \sum_{i \in [n]} \alpha_{ic}^2.$$

The proof is presented in Appendix A.3.

Also, if a pattern c' is a superset of the pattern c, the following relationship holds:

$$x_{ic'} = 1 \;\Rightarrow\; x_{ic} = 1 \quad \forall i,$$

and, conversely

$$x_{ic} = 0 \;\Rightarrow\; x_{ic'} = 0 \quad \forall i.$$

Therefore, the following holds:

$$\sum_{i:\beta_i \tilde{\theta}_i > 0} \alpha_{ic} \tilde{\theta}_i \geq \sum_{i:\beta_i \tilde{\theta}_i > 0} \alpha_{ic'} \tilde{\theta}_i,$$
$$\sum_{i:\beta_i \tilde{\theta}_i < 0} \alpha_{ic} \tilde{\theta}_i \leq \sum_{i:\beta_i \tilde{\theta}_i < 0} \alpha_{ic'} \tilde{\theta}_i.$$

Thus, it is easy to see that the following inequality holds:

$$\mathrm{SSPC}(c) \geq \mathrm{SSPC}(c'). \tag{12}$$

This suggests that the upper bound is becoming tighter for deeper nodes in the tree.

3.3 *Practical considerations*

The safe pattern pruning rule in Theorem 1 depends on a pair of primal feasible solutions $(\tilde{\boldsymbol{w}}, \tilde{b})$ and a dual feasible solution $\tilde{\boldsymbol{\theta}}$. Although the rule can be constructed from any set of solutions as long as they are feasible, the power of the rule depends on the goodness of these solutions. Specifically, the criterion $\mathrm{SSPC}(c)$ depends on the duality gap $P_\lambda(\tilde{\boldsymbol{w}}, \tilde{b}) - D_\lambda(\tilde{\boldsymbol{\theta}})$ which vanishes when these primal and dual solutions are optimal. Roughly speaking, if these solutions are somewhat close to the optimal ones, we could expect the SSP rule to be powerful.

In practice, we need to find a penalty parameter λ based on a model selection technique such as cross-validation. In model selection, a sequence of solutions with various penalty parameters must be trained. Such a sequence of solutions is sometimes referred to as a regularization path [30]. The regularization path of the problem (2) is usually computed with decreasing λ because sparser solutions are obtained for larger λ. Let us write the sequence of λs as $\lambda_0 > \lambda_1 > \ldots > \lambda_K$. When computing such a sequence of solutions, it is reasonable to use the warm-start approach where the previous optimal solution at λ_{k-1} is used as the initial starting point of the next optimization problem at λ_k. In such a situation, we can also use the previous solution at λ_{k-1} for the feasible solution for the safe subsequence pruning rule at λ_k. This strategy is also employed in Nakagawa et al. [22].

In the sparse modeling literature, it is standard practice to start from the largest possible λ at which the primal solution is given as $\boldsymbol{w}^* = \mathbf{0}$ and $b^* = \bar{y}$, where $\bar{y}$ is the sample mean of $\{y_i\}_{i\in[n]}$. The largest λ is given as

$$\lambda_{\max} := \max_{c\in\mathcal{C}} \left| \sum_{i\in[n]} x_{ic}(y_i - \bar{y}) \right|.$$

In order to solve this maximization problem, for a node c and $c' \in \mathcal{C}_{\rm sub}(c)$, we can use the following upper bound

$$\left| \sum_{i\in[n]} x_{ic'}(y_i - \bar{y}) \right| \leq \max \left\{ \sum_{i|y_i-\bar{y}>0} x_{ic}(y_i - \bar{y}), - \sum_{i|y_i-\bar{y}<0} x_{ic}(y_i - \bar{y}) \right\},$$

and this upper bound can be exploited for pruning the search over the tree.

3.4 *Algorithms*

Algorithm 2 shows the proposed method, while Algorithm 3 shows the entire procedure for computing the regularization path by using the SSP rule.

4. Experiments

In order to ascertain the validity of the S3P-classifier described in the previous section, we investigated how the number of nodes traced compared with PrefixSpan is different. In addition, in order to ascertain the possibility of interpretability of the S3P-classifier, we confirmed which pattern was extracted from each data. We used datasets for three animals from animal movement behavior studies, namely ones for streaked shearwater [31], *C. elegans* [32, 33], and crickets [34] for experiments. Table 2 shows the summary of each dataset. All the experiments were performed on an Intel Xeon CPU E5-2620 v4 @ 3.0Ghz server machine with 256-gigabyte main memory. Note that the experiments in this paper use a considerably larger memory than a standard computer machine.

Algorithm 2 Safe Subsequence Pruning

Require: $\{(x_i, y_i)\}_{i\in[n]}$, feasible solution $\{(\tilde{\boldsymbol{w}}, \tilde{b}), \tilde{\boldsymbol{\theta}}\}$
Ensure: $\mathcal{A}$

procedure SafeSubsequencePruning($\tilde{\boldsymbol{w}}, \tilde{b}, \tilde{\boldsymbol{\theta}}$)
  for all singleton patterns $c \in \mathcal{C}$ **do**
    Project(c)
  end for
  return $\mathcal{A}$
end procedure

function Project(c)
  if SSPC(c) < 1 **then**
    return
  end if
  if UB(c) ≥ 1 **then**
    Add c to $\mathcal{A}$
  end if
  scan the projected database $\mathcal{D}_{\boldsymbol{c}}$ once, find every frequently occurring symbol h such that $\boldsymbol{c}$ can be extended to $(\boldsymbol{c} \diamond h)$
  if There is no valid h **then**
    return
  end if
  for $h \in \mathcal{H}$ **do**
    $\boldsymbol{c}' \leftarrow \boldsymbol{c} \diamond h$
    if SSPC(c$'$) ≥ 1 **then**
      Project($\boldsymbol{c}'$)
    end if
  end for
  return
end function

Algorithm 3 Regularization path computation algorithm

Require: $\{(x_i, y_i)\}_{i\in[n]}$, $\{\lambda_k\}_{k\in[K]}$
Ensure: $\{(\boldsymbol{w}^*(\lambda_k), b^*(\lambda_k))\}_{k\in[K]}$ and $\{(\boldsymbol{\theta}^*(\lambda_k)\}_{k\in[K]}$

$\lambda_0 \leftarrow \max_{t\in\mathcal{C}} \left|\sum_{i\in[n]} x_{ic}(y_i - \bar{y})\right|$ and $(\boldsymbol{w}_0, b_0) \leftarrow (\boldsymbol{0}, \bar{y})$
for $k = 1, \ldots, K$ **do**
  Find $\hat{\mathcal{A}}(\lambda_k) \supseteq \mathcal{A}^*(\lambda_k)$ by Safe Subsequence Pruning based on $(\boldsymbol{w}^*(\lambda_{k-1}), b^*(\lambda_{k-1}))$ and $\boldsymbol{\theta}^*(\lambda_{k-1})$ as the primal and dual feasible solutions, respectively.
  Solve the small optimization problems in (6) with $\mathcal{A} = \hat{\mathcal{A}}(\lambda_k)$, and obtain the primal solution $(\boldsymbol{w}^*(\lambda_k), b^*(\lambda_k))$ and the dual solution $\boldsymbol{\theta}^*(\lambda_k)$.
end for

4.1 *Datasets*

4.1.1 C. elegans

The nematode *Caenorhabditis elegans* (*C. elegans*) is a widely-used model organisms. The authors in [33] studied the avoidance behavior of *C.elegans* from the repulsive odor 2-nonanone. The research question discussed in [33] is whether a loss of function via a mutation in a certain gene can change the avoidance behavior. We trained the S3P-classifiers discussed in the previous section to classify the wild-type and mutant strains. Specifically, we compared the avoidance behaviors of the wild-type strains (N2) with each of the three mutant strains called *egl-21*, *egl-*

Table 2. Summary of each dataset

	Dataset name	Label+	Label−	Number of data(+, −)
C. elegans	EGL-21	*egl-21*	N2	72 (36, 36)
	EGL-3	*egl-3*	N2	86 (43, 43)
	DOP-3	*dop-3*	N2	154 (77, 77)
Streaked Shearwater	BIRD	male	female	968 (484, 484)
Cricket	CRICKET	15-kHz	Tone-free	500 (250, 250)

3, and *dop-3*. For each of the three comparisons, we analyzed $n = 72$, 86, and 154 avoidance behaviors. Each avoidance behavior is characterized by three-dimensional symbols representing the moving direction (**F**orward, **B**ackward), two behavioral states (ru**N**, **P**irouette), and the odor concentration change that each worm experienced at the spatio-temporal position (**U**p, **D**own) [2]. Here, the set of symbols $\mathcal{S}$ consists of $2 \times 2 \times 2 = 8$ symbols each of which is a combination of three features such as "Forward Movement", "Pirouette", and "Down", and are denoted as "(F, P, D)".

4.1.2 Streaked Shearwater

Birds have the ability to gather food from locations far away from their nests by using a variety of environmental information. In [31], the authors recorded the trajectories of birds in flight by using GPS loggers. Here, the goal of our analysis is to train a model for classifying differences in the navigation patterns between males and females. We defined a trip to be the movement of a bird which leaves its nest for more than 8 hours. In our analysis, we considered $n = 968 (n_+ = n_- = 484)$ trips. Each trip is represented by a two-dimensional time series of the longitude and the latitude taken every minute. For sequential pattern mining, we extracted the speed (**L**ow or **H**igh), the sea surface temperature (**L**ow, **M**iddle, or **H**igh), and the distance from the coastline (**L**arge, **M**iddle, or **S**mall) at each location along a trajectory, and represent each trip with a sequence of symbols defined by the combination of these three features [3]. Here, the set of symbols $\mathcal{S}$ consists of $2 \times 3 \times 3 = 18$ symbols each of which is a combination of the three features such as "High speed", "Medium seawater temperature", and "Small distance from the coastline", which is denoted as "(H, M, S)".

4.1.3 Crickets

Crickets perform avoidance behaviors in response to air-puff stimuli. Crickets escape from the air-puff stimuli, but when given sound stimuli of a certain frequency prior to the air puff, they perform avoidance behaviors different from their usual ones[34]. In this paper, we analyze the differences in such avoidance behaviors. Here, the goal of our analysis is to train a model for classifying the differences in the behavior patterns of crickets between 15-kHz-tone and tone-free stimuli. In our analysis, we considered the avoidance behaviors of $n = 500 (n_+ = n_- = 250)$ crickets. The cricket's behavior data were recorded in terms of its horizontal coordinates (X, Y) and body axis angle (θ) every 5 ms. Note that the coordinates are X in the lateral direction and Y in the longitudinal direction in reference to the cricket body axis at the start of the experiment, in which the air-puff stimuli were delivered from lateral side corresponding to negative value of X coordinate (Fig. 3). Each avoidance behavior is characterized by three symbols representing the displacement in the X direction (ΔX) (**N**egative, **S**mall-positive, **P**ositive), the displacement

[2]The moving direction is defined as "Forward" if the angle of the movement is in the range of $+90°$ and $-90°$ degrees with a velocity greater than 0.06 mm/s, and "Backward" otherwise. The definitions of the pirouette status is given in [35]. The putative odor concentration change is defined as "Up" when the worm experiences increases in odor concentration, and "Down", otherwise.

[3]The discretization of the speed, the seawater temperature, and the distance from the coastline are defined as follows. The speed is defined as "Low" if it is lower than 10 km/h, and "High" otherwise. The seawater temperature is defined as "Low" if it is lower than 20 degrees, "Medium" if it is in the range from 20 to 25 degrees, and "High" otherwise. The distance from the coastline is defined as "Small" if it is larger than 12,652m, "Medium" if it is in the range from 12,652m to 28,362m, and "Large" otherwise, where the two thresholds were defined based on the empirical 33.3 and 66.6 percentiles.

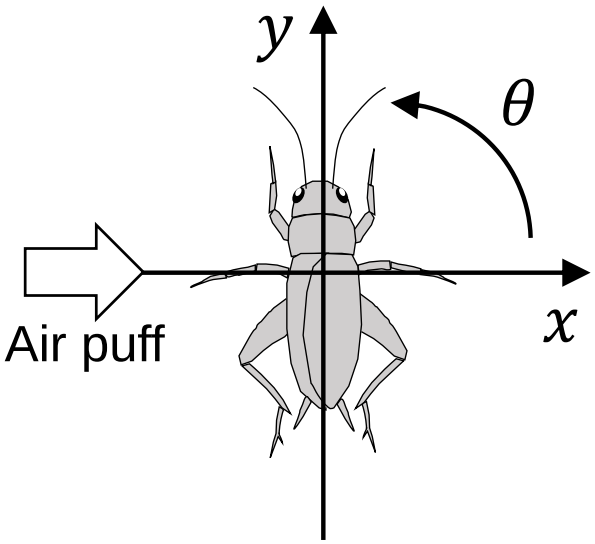

Figure 3. X and Y coordinate axes for the CRICKET dataset

in the Y direction (ΔY) (**N**egative, **Z**ero, **P**ositive), and the displacement in terms of θ ($\Delta\theta$) (**L**arge clockwise, **N**arrowly clockwise, **C**ounterclockwise) [4]. Here, the set of symbols $\mathcal{S}$ consists of $3 \times 3 \times 3 = 27$ symbols each of which is a combination of the three features such as ΔX is "Positive", ΔY is "Negative", and $\Delta\theta$ is "Narrowly clockwise", which are denoted as "(P, N, N)".

4.2 *Parameter Settings*

We considered the regularization path computation scenario described in Section 3.3. Specifically, we computed a sequence of optimal solutions of (2) for a sequence of 100 penalty parameters λ evenly allocated between $\lambda_0 = \lambda_{\max}$ and $0.01\lambda_0$ on the logarithmic scale. Among these λs, those with the best performance from 10-fold cross-validation were selected and used for the analysis.

4.3 *Comparison of computational costs with Prefixspan*

The computational efficiency of the S3P-classifier was verified by comparing the numbers of nodes traced by the S3P-classifier with that by the original PrefixSpan. The number of traced nodes greatly varies depending on maximum sequence length (MSL). Figures 4-8 show the numbers of traced nodes traced. Note that the vertical axis is in logarithmic scale in the figures.

From these figures, it can be seen that the S3P-classifier traced fewer nodes than PrefixSpan in all datasets. The larger the MSL, the wider the difference in the number of nodes, and when MSL is 90, S3P-classifier completed the search with the number of traced nodes less than 0.5% of PrefixSpan. In the BIRD dataset, PrefixSpan could not be completed when MSL is greater than 90. This is because the number of nodes to be held exceeds the main memory size. When using PrefixSpan with a smaller memory machine, it is necessary to further reduce MSL. These results indicate that S3P-classifier is computationally efficient.

Fig. 9 shows the classification performance of S3P-classifier for each MSL as a ratio with respect to the classification performance when MSL is 1. In the dataset of EGL-21, EGL-3, and DOP-3, it can be seen that the classification performance improves as MSL increases. EGL-21

[4]The ΔX is defined as "Negative" if it is lower than 0, "Small-positive" if it is in the range from 0 to 0.19, and "Positive" otherwise. The ΔY is defined as "Negative" if it is lower than -0.15, "Zero" if it is in the range from -0.15 to 0.15, and "Positive" otherwise. The $\Delta\theta$ is defined as "Large clockwise" if it is lower than -1, "Narrowly clockwise" if it is in the range from -1 to 0, and "Counterclockwise" otherwise.

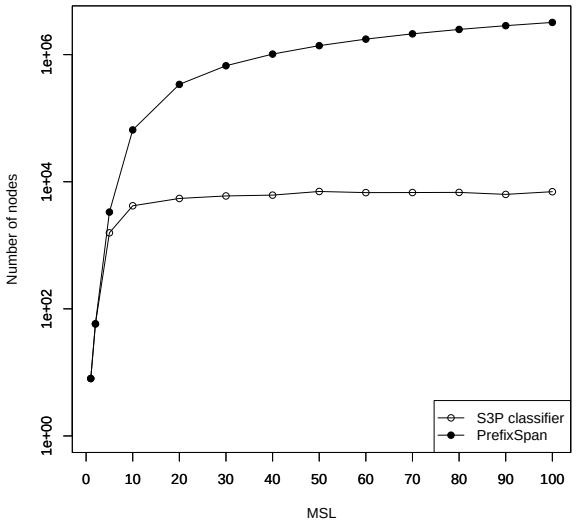

Figure 4. Computational cost for the EGL-21 dataset.

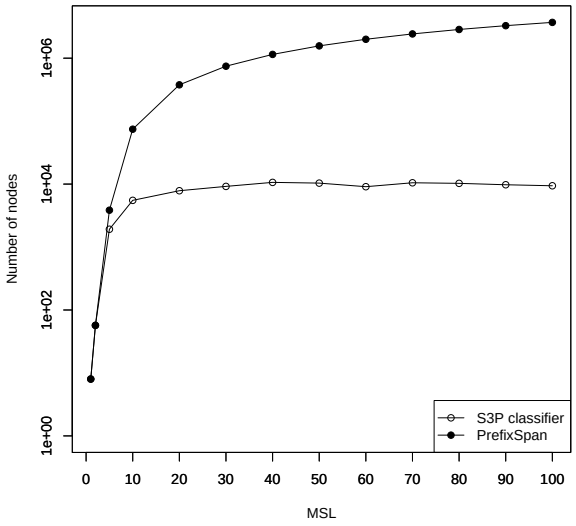

Figure 5. Computational cost for the EGL-3 dataset.

and EGL-3 show the highest classification performance when MSL is 60 and DOP-3 shows the highest classification performance when MSL is 50. Roughly, as MSL is larger, the classification performance improves and MSL contributes to the classification performance improvement of S3P-classifier. We conjecture that the classification performance of the BIRD dataset might improve when we use larger MSL than 100 since the median (IQR) sequence length of the BIRD dataset is 1086 (809.75, 3515.75). On the other hand, we could not expect classification performance improvement of CRICKET dataset for larger MSL because the median (IQR) sequence length is 51 (42, 66.25). A possible remedy for CRICKET dataset is to re-consider the symbolization process (e.g., by changing the abstraction level.)

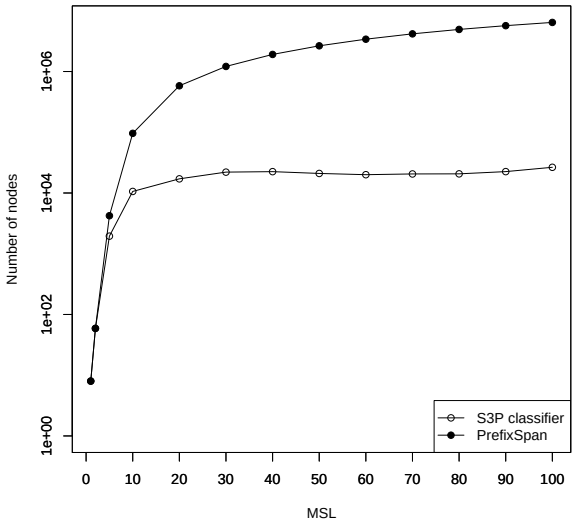

Figure 6. Computational cost for the DOP-3 dataset.

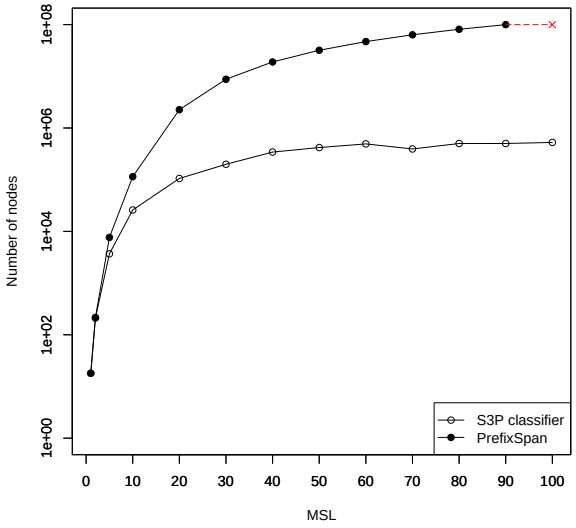

Figure 7. Computational cost for the BIRD dataset. If MSL is 100, PrefixSpan has the huge number of nodes and could not build a tree on memory. The red X mark shows that it could not be completed.

4.4 *Extracted specific patterns*

Tables 3-7 show the sequential patterns whose coefficients in the optimal model are nonzero for each dataset. These tables only show part of the extracted patterns due to the sparse limitation.

Tables 3, 4, and 5 show the results of EGL-21, EGL-3 and DOP-3 dataset, respectively.

The notation "$\langle(\mathrm{F},\mathrm{N},\mathrm{D}) \times 60\rangle$" indicates the pattern in which the symbol "$(\mathrm{F},\mathrm{N},\mathrm{D})$" is repeated 60 times.

The results in the tables clearly indicate that the sequential patterns which appear more frequently in one group than the other are successfully extracted. (Note that sequential patterns whose coefficients are positive/negative suggest that the patterns appear more frequently in the positive/negative groups, respectively.) Table 3 suggests that many patterns containing multiple "$(\mathrm{B},\mathrm{P},\mathrm{U})$" appear frequently in *egl-21*, but rarely appear in N2. On the other hand, many

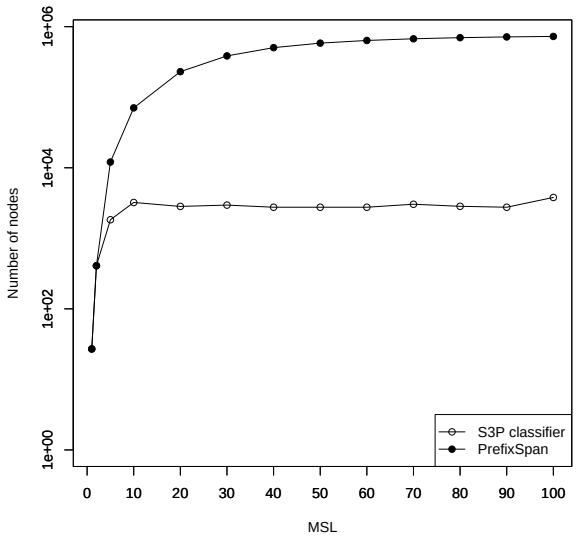

Figure 8. Computational cost for the CRICKET dataset.

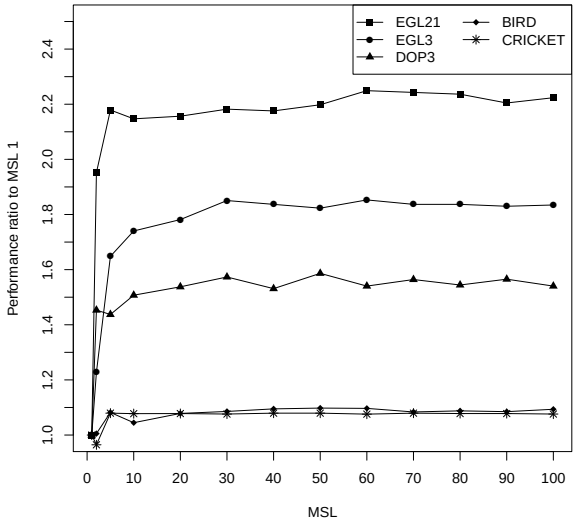

Figure 9. Ratio to MSL 1 in each dataset.

patterns containing multiple "(F, N, D)" frequently appear in N2, but rarely appear in *egl-21*. These results can be interpreted to mean the loss of a function in *egl-21* leads to increased pirouette behavior, which suggests that the *egl-21* fails to directly escape from the odor as in N2. Table 4 shows that *egl-3* has a similar tendency to *egl-21*. However, in Table 4, patterns with a positive w_j also contain "(F, N, D)", meaning that the behavior of *egl-3* is closer to N2 than *egl-21*. These results are reasonable because the functions lost in *egl-21* and *egl-3* are the similar, and the degree of loss of the function is larger in *egl-21* than in *egl-3*. Table 5 shows that many patterns containing multiple "(F, P, D)" appear more frequently in *dop-3*, which is not observed for *egl-21* and *egl-3*, suggesting that the loss of function in *dop-3* leads to different behavior from *egl-21* and *egl-3*.

Table 6 shows the results for the BIRD dataset. From the results, patterns repeating "(H, M, L)" appear more frequently in males, while patterns repeating "(H, H, S)", "(L, H, S)",

Table 3. Specific patterns for the EGL-21 dataset.

Pattern	w_j	support(+)	support(−)
$\langle(\mathrm{B},\mathrm{P},\mathrm{U})\times 10\rangle$	0.240724	35	3
$\langle(\mathrm{B},\mathrm{P},\mathrm{D}),(\mathrm{B},\mathrm{P},\mathrm{U})\times 2,(\mathrm{B},\mathrm{P},\mathrm{D}),(\mathrm{F},\mathrm{P},\mathrm{D})\rangle$	0.161592	20	2
$\langle(\mathrm{B},\mathrm{P},\mathrm{U}),(\mathrm{F},\mathrm{P},\mathrm{U})\times 2,(\mathrm{B},\mathrm{P},\mathrm{U}),(\mathrm{B},\mathrm{P},\mathrm{D})\rangle$	0.15599	30	6
$\langle(\mathrm{B},\mathrm{P},\mathrm{U}),(\mathrm{F},\mathrm{P},\mathrm{U}),(\mathrm{B},\mathrm{P},\mathrm{U})\times 9\rangle$	0.152087	29	2
$\langle(\mathrm{B},\mathrm{P},\mathrm{U})\times 6,(\mathrm{B},\mathrm{P},\mathrm{D}),(\mathrm{B},\mathrm{P},\mathrm{U})\rangle$	0.146988	25	1
$\langle(\mathrm{B},\mathrm{P},\mathrm{D}),(\mathrm{B},\mathrm{P},\mathrm{U})\times 4,(\mathrm{B},\mathrm{P},\mathrm{D})\rangle$	0.141956	27	0
$\langle(\mathrm{B},\mathrm{P},\mathrm{D}),(\mathrm{B},\mathrm{P},\mathrm{U}),(\mathrm{F},\mathrm{P},\mathrm{U}),(\mathrm{B},\mathrm{P},\mathrm{U})\times 2\rangle$	0.141858	20	1
$\langle(\mathrm{B},\mathrm{P},\mathrm{U})\times 9\rangle$	0.133809	36	4
$\langle(\mathrm{F},\mathrm{P},\mathrm{U})\times 2,(\mathrm{B},\mathrm{P},\mathrm{U}),(\mathrm{B},\mathrm{P},\mathrm{D}),(\mathrm{B},\mathrm{P},\mathrm{U})\rangle$	0.120514	23	5
$\langle(\mathrm{B},\mathrm{P},\mathrm{U})\times 13\rangle$	0.119841	31	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\langle(\mathrm{F},\mathrm{N},\mathrm{D})\times 59\rangle$	-0.0263455	1	29
$\langle(\mathrm{F},\mathrm{N},\mathrm{U}),(\mathrm{F},\mathrm{N},\mathrm{D})\times 21\rangle$	-0.0265882	0	28
$\langle(\mathrm{F},\mathrm{N},\mathrm{D})\times 38,(\mathrm{B},\mathrm{N},\mathrm{D})\rangle$	-0.0391795	5	25
$\langle(\mathrm{F},\mathrm{N},\mathrm{D})\times 34,(\mathrm{B},\mathrm{N},\mathrm{D})\rangle$	-0.0442307	6	27
$\langle(\mathrm{F},\mathrm{N},\mathrm{D})\times 33,(\mathrm{B},\mathrm{N},\mathrm{D})\rangle$	-0.0543381	6	28
$\langle(\mathrm{F},\mathrm{P},\mathrm{D}),(\mathrm{F},\mathrm{N},\mathrm{D})\times 37\rangle$	-0.116666	4	26
$\langle(\mathrm{F},\mathrm{N},\mathrm{D}),(\mathrm{B},\mathrm{P},\mathrm{D})\rangle$	-0.140089	5	30
$\langle(\mathrm{F},\mathrm{N},\mathrm{D})\times 58\rangle$	-0.153787	1	30
$\langle(\mathrm{F},\mathrm{N},\mathrm{U}),(\mathrm{F},\mathrm{N},\mathrm{D})\times 16\rangle$	-0.211889	0	29
$\langle(\mathrm{F},\mathrm{N},\mathrm{D})\times 52\rangle$	-0.277221	2	33

Table 4. Specific patterns for the EGL-3 dataset.

Pattern	w_j	support(+)	support(−)
$\langle(\mathrm{B},\mathrm{P},\mathrm{D}),(\mathrm{F},\mathrm{P},\mathrm{D}),(\mathrm{B},\mathrm{P},\mathrm{D}),(\mathrm{B},\mathrm{P},\mathrm{U})\rangle$	0.0837365	32	11
$\langle(\mathrm{B},\mathrm{P},\mathrm{D}),(\mathrm{B},\mathrm{P},\mathrm{U}),(\mathrm{F},\mathrm{P},\mathrm{U})\times 2,(\mathrm{B},\mathrm{P},\mathrm{D})\rangle$	0.0754945	28	11
$\langle(\mathrm{B},\mathrm{P},\mathrm{U})\times 9\rangle$	0.0516311	28	6
$\langle(\mathrm{F},\mathrm{N},\mathrm{D}),(\mathrm{B},\mathrm{P},\mathrm{U})\times 2\rangle$	0.0496707	25	5
$\langle(\mathrm{F},\mathrm{N},\mathrm{D})\times 6,(\mathrm{B},\mathrm{P},\mathrm{U})\times 2\rangle$	0.017552	18	3
$\langle(\mathrm{B},\mathrm{P},\mathrm{U})\times 3,(\mathrm{F},\mathrm{P},\mathrm{U}),(\mathrm{B},\mathrm{P},\mathrm{U})\times 3\rangle$	0.0136679	30	11
$\langle(\mathrm{B},\mathrm{P},\mathrm{U})\times 2,(\mathrm{F},\mathrm{P},\mathrm{U}),(\mathrm{B},\mathrm{P},\mathrm{U})\times 2\rangle$	0.00762181	36	17
$\langle(\mathrm{B},\mathrm{P},\mathrm{U}),(\mathrm{F},\mathrm{P},\mathrm{U}),(\mathrm{B},\mathrm{P},\mathrm{U}),(\mathrm{B},\mathrm{P},\mathrm{D})\rangle$	0.000413603	35	15
$\langle(\mathrm{F},\mathrm{P},\mathrm{U})\times 8\rangle$	-0.00182683	7	26
$\langle(\mathrm{B},\mathrm{P},\mathrm{D})\times 3,(\mathrm{F},\mathrm{N},\mathrm{D})\times 25\rangle$	-0.0023657	5	23
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\langle(\mathrm{B},\mathrm{P},\mathrm{D})\times 3,(\mathrm{F},\mathrm{N},\mathrm{D})\times 29\rangle$	-0.112826	3	22
$\langle(\mathrm{F},\mathrm{N},\mathrm{U}),(\mathrm{F},\mathrm{N},\mathrm{D})\times 21\rangle$	-0.132935	9	34
$\langle(\mathrm{F},\mathrm{N},\mathrm{D})\times 13,(\mathrm{B},\mathrm{N},\mathrm{D})\times 2\rangle$	-0.138766	3	21
$\langle(\mathrm{F},\mathrm{P},\mathrm{U})\times 5,(\mathrm{B},\mathrm{P},\mathrm{U})\rangle$	-0.151166	12	34
$\langle(\mathrm{F},\mathrm{N},\mathrm{D})\times 30,(\mathrm{B},\mathrm{N},\mathrm{D})\rangle$	-0.183145	9	36
$\langle(\mathrm{F},\mathrm{N},\mathrm{D})\times 52\rangle$	-0.19513	14	40
$\langle(\mathrm{F},\mathrm{P},\mathrm{U})\times 9\rangle$	-0.198477	1	21
$\langle(\mathrm{F},\mathrm{N},\mathrm{D})\times 23,(\mathrm{B},\mathrm{N},\mathrm{D}),(\mathrm{F},\mathrm{N},\mathrm{D})\rangle$	-0.273485	6	33
$\langle(\mathrm{F},\mathrm{N},\mathrm{D}),(\mathrm{B},\mathrm{P},\mathrm{U}),(\mathrm{F},\mathrm{P},\mathrm{U})\times 3\rangle$	-0.377975	3	25
$\langle(\mathrm{F},\mathrm{P},\mathrm{D})\times 9\rangle$	-0.552839	4	26

and "$(\mathrm{L},\mathrm{M},\mathrm{S})$" appear more frequently in females. These results can be interpreted that males tend to travel further from the coastline than females.

Table 7 shows the results for the CRICKET dataset. These results show that the pattern "$(\mathrm{P},\mathrm{N},\mathrm{N})\times 3$" appear more frequently at 15 kHz, meaning that the crickets moved diagonally backward with small turn against the air-puff stimuli. Repeating of this sequential pattern would result in the backward bias of the escape direction that is induced by preceding sound stimuli, as reported in our previous study [34].

Table 5. Specific patterns for the DOP-3 dataset.

Pattern	w_j	support(+)	support(−)
$\langle(\mathrm{F},\mathrm{P},\mathrm{D})\times 14\rangle$	0.268911	24	1
$\langle(\mathrm{F},\mathrm{P},\mathrm{D})\times 10\rangle$	0.185818	62	35
$\langle(\mathrm{B},\mathrm{P},\mathrm{U})\times 5,(\mathrm{F},\mathrm{P},\mathrm{U})\times 2\rangle$	0.164925	43	19
$\langle(\mathrm{F},\mathrm{P},\mathrm{U}),(\mathrm{F},\mathrm{P},\mathrm{D})\times 3\rangle$	0.114146	61	40
$\langle(\mathrm{F},\mathrm{P},\mathrm{U})\times 9,(\mathrm{B},\mathrm{P},\mathrm{U})\rangle$	0.0996034	36	16
$\langle(\mathrm{F},\mathrm{P},\mathrm{D})\times 7,(\mathrm{F},\mathrm{P},\mathrm{U})\rangle$	0.0732906	36	13
$\langle(\mathrm{F},\mathrm{P},\mathrm{D}),(\mathrm{F},\mathrm{P},\mathrm{U})\times 4,(\mathrm{B},\mathrm{P},\mathrm{D})\rangle$	0.0452207	22	11
$\langle(\mathrm{F},\mathrm{P},\mathrm{D})\times 5,(\mathrm{B},\mathrm{P},\mathrm{D}),(\mathrm{F},\mathrm{P},\mathrm{D})\rangle$	0.0429675	44	22
$\langle(\mathrm{F},\mathrm{P},\mathrm{U}),(\mathrm{F},\mathrm{P},\mathrm{D})\times 7\rangle$	0.0351105	28	9
$\langle(\mathrm{F},\mathrm{P},\mathrm{U})\times 7,(\mathrm{B},\mathrm{P},\mathrm{D}),(\mathrm{F},\mathrm{P},\mathrm{D})\times 4\rangle$	0.0286394	29	15
⋮	⋮	⋮	⋮
$\langle(\mathrm{F},\mathrm{N},\mathrm{D})\times 21,(\mathrm{F},\mathrm{N},\mathrm{U}),(\mathrm{F},\mathrm{N},\mathrm{D})\rangle$	-0.0254462	23	39
$\langle(\mathrm{B},\mathrm{P},\mathrm{D})\times 3,(\mathrm{F},\mathrm{N},\mathrm{D})\rangle$	-0.0563884	22	46
$\langle(\mathrm{B},\mathrm{P},\mathrm{D})\times 4,(\mathrm{F},\mathrm{N},\mathrm{D})\rangle$	-0.0608795	10	32
$\langle(\mathrm{F},\mathrm{P},\mathrm{D}),(\mathrm{F},\mathrm{N},\mathrm{D})\times 47\rangle$	-0.082303	12	40
$\langle(\mathrm{F},\mathrm{P},\mathrm{U}),(\mathrm{B},\mathrm{P},\mathrm{D})\times 4\rangle$	-0.100007	31	50
$\langle(\mathrm{F},\mathrm{N},\mathrm{D}),(\mathrm{B},\mathrm{P},\mathrm{D})\rangle$	-0.15073	26	59
$\langle(\mathrm{B},\mathrm{P},\mathrm{D}),(\mathrm{F},\mathrm{N},\mathrm{D})\times 30\rangle$	-0.171908	20	51
$\langle(\mathrm{F},\mathrm{N},\mathrm{D})\times 59\rangle$	-0.199281	20	56
$\langle(\mathrm{F},\mathrm{N},\mathrm{D})\times 2,(\mathrm{B},\mathrm{P},\mathrm{D})\rangle$	-0.225817	21	55
$\langle(\mathrm{F},\mathrm{N},\mathrm{D})\times 48\rangle$	-0.234486	35	70

Table 6. Specific patterns for the BIRD dataset.

Pattern	w_j	support(+)	support(−)
$\langle(\mathrm{H},\mathrm{L},\mathrm{S})\times 8\rangle$	0.265474	54	6
$\langle(\mathrm{H},\mathrm{M},\mathrm{L})\times 47\rangle$	0.201201	148	52
$\langle(\mathrm{L},\mathrm{M},\mathrm{L})\times 2,(\mathrm{H},\mathrm{M},\mathrm{L})\times 33\rangle$	0.148875	139	46
$\langle(\mathrm{L},\mathrm{H},\mathrm{L})\times 11,(\mathrm{H},\mathrm{H},\mathrm{L})\times 2,(\mathrm{L},\mathrm{H},\mathrm{L})\times 2\rangle$	0.114139	86	45
$\langle(\mathrm{H},\mathrm{M},\mathrm{L})\times 11,(\mathrm{H},\mathrm{M},\mathrm{M})\times 18\rangle$	0.108663	122	56
$\langle(\mathrm{H},\mathrm{H},\mathrm{M})\times 19,(\mathrm{H},\mathrm{H},\mathrm{L})\times 20\rangle$	0.100756	84	38
$\langle(\mathrm{H},\mathrm{H},\mathrm{S})\rangle$	0.0993539	368	348
$\langle(\mathrm{H},\mathrm{M},\mathrm{L})\times 29\rangle$	0.0611184	177	77
$\langle(\mathrm{L},\mathrm{M},\mathrm{M})\times 40,(\mathrm{H},\mathrm{M},\mathrm{L})\times 4\rangle$	0.0517693	129	103
$\langle(\mathrm{H},\mathrm{H},\mathrm{M})\times 19,(\mathrm{H},\mathrm{H},\mathrm{L})\times 16\rangle$	0.0452597	93	46
⋮	⋮	⋮	⋮
$\langle(\mathrm{L},\mathrm{H},\mathrm{S}),(\mathrm{H},\mathrm{H},\mathrm{S}),(\mathrm{L},\mathrm{H},\mathrm{S})\times 4,(\mathrm{H},\mathrm{H},\mathrm{S})\rangle$	-0.0615068	97	166
$\langle(\mathrm{H},\mathrm{H},\mathrm{S})\times 29\rangle$	-0.0635537	145	193
$\langle(\mathrm{L},\mathrm{H},\mathrm{S}),(\mathrm{H},\mathrm{H},\mathrm{S}),(\mathrm{L},\mathrm{H},\mathrm{S})\times 7,(\mathrm{H},\mathrm{H},\mathrm{S})\rangle$	-0.0695386	65	126
$\langle(\mathrm{L},\mathrm{M},\mathrm{M})\times 32,(\mathrm{H},\mathrm{M},\mathrm{M}),(\mathrm{L},\mathrm{M},\mathrm{M})\times 16\rangle$	-0.0706738	63	92
$\langle(\mathrm{L},\mathrm{H},\mathrm{S})\times 3,(\mathrm{H},\mathrm{H},\mathrm{S}),(\mathrm{L},\mathrm{H},\mathrm{S})\times 17\rangle$	-0.0738069	151	224
$\langle(\mathrm{L},\mathrm{M},\mathrm{M})\times 3,(\mathrm{H},\mathrm{M},\mathrm{M}),(\mathrm{L},\mathrm{M},\mathrm{M}),(\mathrm{H},\mathrm{M},\mathrm{M}),(\mathrm{L},\mathrm{M},\mathrm{M})\rangle$	-0.0832406	63	94
$\langle(\mathrm{L},\mathrm{M},\mathrm{S}),(\mathrm{H},\mathrm{M},\mathrm{S}),(\mathrm{L},\mathrm{M},\mathrm{S})\times 19\rangle$	-0.0851661	103	143
$\langle(\mathrm{L},\mathrm{M},\mathrm{S})\times 10,(\mathrm{H},\mathrm{M},\mathrm{S})\times 2,(\mathrm{L},\mathrm{M},\mathrm{S}),(\mathrm{H},\mathrm{M},\mathrm{S})\times 2\rangle$	-0.0887395	13	43
$\langle(\mathrm{L},\mathrm{H},\mathrm{S}),(\mathrm{H},\mathrm{H},\mathrm{S})\times 15\rangle$	-0.151165	185	256
$\langle(\mathrm{L},\mathrm{H},\mathrm{M})\times 10,(\mathrm{H},\mathrm{H},\mathrm{M}),(\mathrm{L},\mathrm{H},\mathrm{M})\times 11,(\mathrm{H},\mathrm{H},\mathrm{M})\rangle$	-0.195478	12	48

Table 7. Specific patterns for the CRICKET dataset.

Pattern	w_j	support(+)	support(−)
$\langle(\mathrm{P},\mathrm{N},\mathrm{N})\times 3\rangle$	0.125176	129	80

5. Conclusion

In this paper, we introduced a sparse subsequence pattern-based classifier (S3P-classifier) for comparing animal movement behavior in two groups of animals. The main advantage of the S3P-classifier is its interpretability. By examining the sequential patterns whose corresponding coefficients are positive/negative, we can extract the movement behavior specific to either of the two classes.

However, fitting S3P-classifiers is computationally challenging since the number of all possible

sequential patterns is extremely large. In order to overcome this difficulty, we developed a novel algorithm by combining the safe screening and sequential mining techniques. This allows us to screen out multiple irrelevant sequential patterns efficiently.

We demonstrated the usefulness of the S3P-classifier by applying it to animal movement behavior datasets for three animals. By examining the sequential patterns which are active in the optimal models, we could obtain new biological knowledge about movement behaviors specific to one of the two groups of animals.

Acknowledgement

This work was partially supported by MEXT KAKENHI (18K18010, 17H04694, 17H00758, 16H06544, 16H06538), JST PRESTO (JPMJPR15N2), JST CREST (JPMJCR1302, JPMJCR1502), the Advanced Intelligence Project of the RIKEN Center, and the JST initiative for promoting materials research by information integration.

References

[1] El Ghaoui L, Viallon V, Rabbani T. Safe feature elimination for the lasso and sparse supervised learning problems. Pacific Journal of Optimization. 2012;8(4):667–698.
[2] Xiang ZJ, Xu H, Ramadge PJ. Learning sparse representations of high dimensional data on large scale dictionaries. In: Advances in neural information processing systems. 2011. p. 900–908.
[3] Wang J, Zhou J, Wonka P, Ye J. Lasso screening rules via dual polytope projection. In: Advances in neural information processing systems. 2013. p. 1070–1078.
[4] Bonnefoy A, Emiya V, Ralaivola L, Gribonval R. A dynamic screening principle for the lasso. In: Signal processing conference (eusipco), 2014 proceedings of the 22nd european. IEEE. 2014. p. 6–10.
[5] Liu J, Zhao Z, Wang J, Ye J. Safe Screening with Variational Inequalities and Its Application to Lasso. In: Proceedings of the 31st international conference on machine learning. 2014.
[6] Wang J, Zhou J, Liu J, Wonka P, Ye J. A safe screening rule for sparse logistic regression. In: Advances in neural information processing systems. 2014. p. 1053–1061.
[7] Xiang ZJ, Wang Y, Ramadge PJ. Screening tests for lasso problems. arXiv preprint arXiv:14054897. 2014;.
[8] Fercoq O, Gramfort A, Salmon J. Mind the duality gap: safer rules for the lasso. In: Proceedings of the 32nd international conference on machine learning. 2015. p. 333–342.
[9] Ndiaye E, Fercoq O, Gramfort A, Salmon J. Gap safe screening rules for sparse multi-task and multi-class models. In: Advances in neural information processing systems. 2015. p. 811–819.
[10] Han J, Pei J, Mortazavi-Asl B, Pinto H, Chen Q, Dayal U, Hsu M. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In: proceedings of the 17th international conference on data engineering. 2001. p. 215–224.
[11] Wang J, Han J, Li C. Frequent closed sequence mining without candidate maintenance. IEEE Transactions on Knowledge and Data Engineering. 2007;19(8):1042–1056.
[12] Fu Tc. A review on time series data mining. Engineering Applications of Artificial Intelligence. 2011; 24(1):164–181.
[13] Srikant R, Agrawal R. Mining sequential patterns: Generalizations and performance improvements. In: International conference on extending database technology. Springer. 1996. p. 1–17.
[14] Fournier-Viger P, Gomariz A, Campos M, Thomas R. Fast vertical mining of sequential patterns using co-occurrence information. In: Pacific-asia conference on knowledge discovery and data mining. Springer. 2014. p. 40–52.
[15] Zaki MJ. Spade: An efficient algorithm for mining frequent sequences. Machine learning. 2001;42(1-2):31–60.
[16] Ayres J, Flannick J, Gehrke J, Yiu T. Sequential pattern mining using a bitmap representation. In: Proceedings of the eighth acm sigkdd international conference on knowledge discovery and data mining. ACM. 2002. p. 429–435.

[17] Yang Z, Kitsuregawa M. Lapin-spam: An improved algorithm for mining sequential pattern. In: Data engineering workshops, 2005. 21st international conference on. IEEE. 2005. p. 1222–1222.
[18] Gouda K, Hassaan M, Zaki MJ. Prism: An effective approach for frequent sequence mining via prime-block encoding. Journal of Computer and System Sciences. 2010;76(1):88–102.
[19] Salvemini E, Fumarola F, Malerba D, Han J. Fast sequence mining based on sparse id-lists. In: International symposium on methodologies for intelligent systems. Springer. 2011. p. 316–325.
[20] Aggarwal CC, Han J. Frequent pattern mining. Springer. 2014.
[21] Fournier-Viger P, Lin JCW, Kiran RU, Koh YS, Thomas R. A survey of sequential pattern mining. Data Science and Pattern Recognition. 2017;1(1):54–77.
[22] Nakagawa K, Suzumura S, Karasuyama M, Tsuda K, Takeuchi I. Safe pattern pruning: An efficient approach for predictive pattern mining. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016.
[23] Sakuma T, Nishi K, Yamazaki SJ, Kimura KD, Matsumoto S, Yoda K, Takeuchi I. Finding discriminative animal behaviors from sequential bio-logging trajectory data. In: Hci international. Vol. 2018. 2018. p. in press.
[24] Lin J, Keogh E, Wei L, Lonardi S. Experiencing sax: a novel symbolic representation of time series. Data Mining and knowledge discovery. 2007;15(2):107–144.
[25] Shieh J, Keogh E. isax: disk-aware mining and indexing of massive time series datasets. Data Mining and Knowledge Discovery. 2009;19(1):24–57.
[26] Camerra A, Palpanas T, Shieh J, Keogh E. isax 2.0: Indexing and mining one billion time series. In: Data mining (icdm), 2010 ieee 10th international conference on. IEEE. 2010. p. 58–67.
[27] Pham ND, Le QL, Dang TK. Two novel adaptive symbolic representations for similarity search in time series databases. In: Web conference (apweb), 2010 12th international asia-pacific. IEEE. 2010. p. 181–187.
[28] Rish I, Grabarnik G. Sparse modeling: theory, algorithms, and applications. CRC press. 2014.
[29] Tibshirani R. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B (Methodological). 1996;:267–288.
[30] Park MY, Hastie T. L1-regularization path algorithm for generalized linear models. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2007;69(4):659–677.
[31] Matsumoto S, Yamamoto T, Yamamoto M, Zavalaga CB, Yoda K. Sex-related differences in the foraging movement of streaked shearwaters calonectris leucomelas breeding on awashima island in the sea of japan. Ornithological Science. 2017;16(1):23–32.
[32] Yamazoe-Umemoto A, Fujita K, Iino Y, Iwasaki Y, Kimura KD. Modulation of different behavioral components by neuropeptide and dopamine signalings in non-associative odor learning of *Caenorhabditis elegans*. Neuroscience Research. 2015;99:22–33.
[33] Kimura KD, Fujita K, Katsura I. Enhancement of odor avoidance regulated by dopamine signaling in *Caenorhabditis elegans*. J Neurosci. 2010;30:16365–16375.
[34] Fukutomi M, Ogawa H. Crickets alter wind-elicited escape strategies depending on acoustic context. Scientific reports. 2017;7(1):15158.
[35] Pierce-Shimomura JT, Morse TM, Lockery SR. The fundamental role of pirouettes in *Caenorhabditis elegans* chemotaxis. Journal of Neuroscience. 1999;19:9557–9569.
[36] Boyd S, Vandenberghe L. Convex optimization. Cambridge university press. 2004.

Appendix A. Proofs

A.1 *Proof of Lemma 1*

Proof. Based on convex optimization theory (see, e.g., [36]), the KKT optimality condition of the primal problem (2) and the dual problem (9) is written as

$$\sum_{i=1}^{n} \alpha_{ij} \theta_i^* \in \begin{cases} \mathrm{sign}(w_j^*) & \text{if } w_j^* \neq 0, \\ [-1, +1] & \text{if } w_j^* = 0, \end{cases}$$

This suggests that

$$\left|\sum_{i=1}^{n} \alpha_{ij}\theta_i^*\right| < 1 \;\Rightarrow\; w_j^* = 0.$$

□

A.2 *Proof of Lemma 3*

Proof. Let $\boldsymbol{\alpha}_{:,j} := [\alpha_{1j}, \ldots, \alpha_{nj}]^\top$. First, note that the objective part of the optimization problem (10) is rewritten as

$$\begin{aligned} & \max_{\boldsymbol{\theta}} \quad \left|\boldsymbol{\alpha}_{:,j}^\top \boldsymbol{\theta}\right| \\ \Leftrightarrow \quad & \max_{\boldsymbol{\theta}} \max\left\{\boldsymbol{\alpha}_{:,j}^\top \boldsymbol{\theta}, -\boldsymbol{\alpha}_{:,j}^\top \boldsymbol{\theta}\right\} \\ \Leftrightarrow \quad & \max\left\{-\min_{\boldsymbol{\theta}}(-\boldsymbol{\alpha}_{:,j})^\top \boldsymbol{\theta}, -\min_{\boldsymbol{\theta}} \boldsymbol{\alpha}_{:,j}^\top \boldsymbol{\theta}\right\} \end{aligned} \tag{A1}$$

Thus, we consider the following convex optimization problem:

$$\min_{\boldsymbol{\theta}} \; \boldsymbol{\alpha}_{:,j}^\top \boldsymbol{\theta} \text{ s.t. } \|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2^2 \le r_\lambda^2, \boldsymbol{\beta}^\top \boldsymbol{\theta} = 0. \tag{A2}$$

Let us define the Lagrange function

$$L(\boldsymbol{\theta}, \xi, \eta) = \boldsymbol{\alpha}_{:,j}^\top \boldsymbol{\theta} + \xi(\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2^2 - r_\lambda^2) + \eta \boldsymbol{\beta}^\top \boldsymbol{\theta},$$

and then the optimization problem (A2) is written as

$$\min_{\boldsymbol{\theta}} \max_{\xi \ge 0, \eta} L(\boldsymbol{\theta}, \xi, \eta). \tag{A3}$$

The KKT optimality conditions are summarized as

$$\xi > 0, \tag{A4a}$$

$$\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2^2 - r_\lambda^2 \le 0, \tag{A4b}$$

$$\boldsymbol{\beta}^\top \boldsymbol{\theta} = 0, \tag{A4c}$$

$$\xi(\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2^2 - r_\lambda^2) = 0, \tag{A4d}$$

where $\xi > 0$ because the problem does not have a minimum value when $\xi = 0$. Differentiating the Lagrange function w.r.t. $\boldsymbol{\theta}$ and using the fact that the result should be zero,

$$\boldsymbol{\theta} = \tilde{\boldsymbol{\theta}} - \frac{1}{2\xi}(\boldsymbol{\alpha}_{:,j} + \eta\boldsymbol{\beta}). \tag{A5}$$

Substituting (A5) into (A3),

$$\max_{\xi > 0, \eta} -\frac{1}{4\xi}\|\boldsymbol{\alpha}_{:,j} + \eta\boldsymbol{\beta}\|_2^2 + (\boldsymbol{\alpha}_{:,j} + \eta\boldsymbol{\beta})^\top \tilde{\boldsymbol{\theta}} - \xi r_\lambda^2.$$

Since the objective function is a quadratic concave function w.r.t. η, we obtain the following by considering the condition (A4c):

$$\eta = -\frac{\boldsymbol{\alpha}_{:,j}^\top \boldsymbol{\beta}}{\|\boldsymbol{\beta}\|_2^2}.$$

By substituting this into (A5),

$$\boldsymbol{\theta} = \tilde{\boldsymbol{\theta}} - \frac{1}{2\xi}\left(\boldsymbol{\alpha}_{:,t} - \frac{\boldsymbol{\alpha}_{:,j}^\top \boldsymbol{\beta}}{\|\boldsymbol{\beta}\|_2^2}\boldsymbol{\beta}\right). \tag{A6}$$

Since $\xi > 0$ and (A4d) indicates $\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2^2 - r_\lambda^2 = 0$, by substituting (A6) into this equality,

$$\xi = \frac{1}{2\|\boldsymbol{\beta}\|_2 r_\lambda}\sqrt{\|\boldsymbol{\alpha}_{:,j}\|_2^2\|\boldsymbol{\beta}\|_2^2 - (\boldsymbol{\alpha}_{:,j}^\top \boldsymbol{\beta})^2}.$$

Then, from (A6), the solution of (A2) is given as

$$\boldsymbol{\theta} = \tilde{\boldsymbol{\theta}} - \frac{\|\boldsymbol{\beta}\|_2 r_\lambda}{\sqrt{\|\boldsymbol{\alpha}_{:,j}\|_2^2\|\boldsymbol{\beta}\|_2^2 - (\boldsymbol{\alpha}_{:,j}^\top \boldsymbol{\beta})^2}}\left(\boldsymbol{\alpha}_{:,j} - \frac{\boldsymbol{\alpha}_{:,j}^\top \boldsymbol{\beta}}{\|\boldsymbol{\beta}\|_2^2}\boldsymbol{\beta}\right),$$

and the minimum objective function value of (A2) is

$$\boldsymbol{\alpha}_{:,j}^\top \tilde{\boldsymbol{\theta}} - r_\lambda\sqrt{\|\boldsymbol{\alpha}_{:,j}\|_2^2 - \frac{(\boldsymbol{\alpha}_{:,j}^\top \boldsymbol{\beta})^2}{\|\boldsymbol{\beta}\|_2^2}}. \tag{A7}$$

Then, substituting (A7) into (A1), the optimal objective value of (10) is given as

$$\left|\boldsymbol{\alpha}_{:,j}^\top \tilde{\boldsymbol{\theta}}\right| + r_\lambda\sqrt{\|\boldsymbol{\alpha}_{:,j}\|_2^2 - \frac{(\boldsymbol{\alpha}_{:,j}^\top \boldsymbol{\beta})^2}{\|\boldsymbol{\beta}\|_2^2}}.$$

□

A.3 *Proof of Theorem 1*

In the next lemma, we show that $\mathrm{SSPC}(c) \geq \mathrm{UB}(c')$ for $\forall c' \in \mathcal{C}_{\mathrm{sub}}(c)$, i.e., $\mathrm{SSPC}(c)$ in Theorem 1 is an upper bound of $\mathrm{UB}(c')$, which enables us to efficiently prune subtrees during the tree traversing process.

Lemma 4. *For any $c' \in \mathcal{C}_{\mathrm{sub}}(c)$,*

$$\begin{aligned}\mathrm{UB}(c') = \left|\sum_{i\in[n]} \alpha_{ic'}\tilde{\theta}_i\right| + r_\lambda\sqrt{\sum_{i\in[n]}\alpha_{ic'}^2 - \frac{(\sum_{i\in[n]}\alpha_{ic'}\beta_i)^2}{\|\boldsymbol{\beta}\|_2^2}} \\ \leq u_t + r_\lambda\sqrt{v_c} = \mathrm{SSPC}(c).\end{aligned}$$

Finally, by combining Lemmas 1, 2, 3, and 4, we can prove Theorem 1.
Proof of Theorem 1.

Proof. From Lemmas 2, 3, and 4,

$$\left| \sum_{i \in [n]} \alpha_{ic'} \theta_i^* \right| \leq \mathrm{UB}(c') \leq \mathrm{SSPC}(c), \quad \forall c' \in \mathcal{C}_{\mathrm{sub}}(c). \tag{A8}$$

From Lemma 1 and (A8),

$$\mathrm{SSPC}(c) < 1 \;\Rightarrow\; w_{c'}^* = 0, \quad \forall c' \in \mathcal{C}_{\mathrm{sub}}(c).$$

□