

DISTANCES FOR COMPARING MULTISSETS AND SEQUENCES

BY GEORGE BOLT^{1,a}, SIMÓN LUNAGÓMEZ^{2,c} AND CHRISTOPHER NEMETH^{1,b} 

¹Lancaster University, ^ag.bolt@lancaster.ac.uk; ^bc.nemeth@lancaster.ac.uk

²Instituto Tecnológico Autónomo de México (ITAM), ^csimon.lunagomez@itam.mx

Measuring the distance between data points is fundamental to many statistical techniques, such as dimension reduction or clustering algorithms. However, improvements in data collection technologies has led to a growing versatility of structured data for which standard distance measures are inapplicable. In this paper, we consider the problem of measuring the distance between sequences and multisets of points lying within a metric space, motivated by the analysis of an in-play football data set. Drawing on the wider literature, including that of time series analysis and optimal transport, we discuss various distances which are available in such an instance. For each distance, we state and prove theoretical properties, proposing possible extensions where they fail. Finally, via an example analysis of the in-play football data, we illustrate the usefulness of these distances in practice.

1. Introduction. Distance measures represent a versatile tool for the practicing data analyst. Once a distance has been specified, an array of subsequent methodologies immediately become available. These include clustering algorithms such as hierarchical clustering (Izenman, 2008, Sec. 12.3) and DBSCAN (Ester et al., 1996), placing data points into groups; dimension reduction or embedding techniques, such as multidimensional scaling (MDS) (Izenman, 2008, Ch. 13) and UMAP (McInnes, Healy and Melville, 2018; Becht et al., 2019), facilitating data visualisation; or prediction algorithms such as k-nearest neighbours regression (Hastie et al., 2009, Sec. 13.3).

However, with improvements in data collection comes an increasingly diverse array of structured data for which standard distance measures are unsuitable. This motivates consideration of distances tailored to fit the objects of focus. Examples include graph distances (Donnat and Holmes, 2018), appearing frequently in the network data analysis literature, or distances between ranks (Kumar and Vassilvitskii, 2010), which often appear in the context of preference learning.

In this paper, we consider the problem of eliciting distances between sequences and multisets. In the most general sense, a sequence is an enumerated collection of objects within some underlying space, with a multiset being the un-ordered analogue of a sequence. An intuitive example is a text document. Naturally, this can be seen as a sequence of words. However, it can also be seen as a multiset of words, or what is referred to by some as a ‘bag-of-words’ (Kusner et al., 2015), wherein two documents equal up to a permutation of word order would be considered one and the same. Other examples of data interpretable in this manner are

- Temporal networks, for example, in the analysis of Donnat and Holmes (2018) biological measurements were encoded via a graph for a given patient through a longitudinal study;
- User interactions within online platforms. For example, the Foursquare data set (Yang et al., 2015) records users checking into different venues throughout the day, e.g. cinemas, cafes, sports venues etc., which leads to a sequence of sequences for each user, with the inner sequence for a user representing one day of their venue check-ins;
- Historical purchases, where the purchase history of a single customer could be represented as a sequence or multiset of orders, with each order encoded as set of products. Such data often appears in the market basket analysis literature (Raeder and Chawla, 2011).

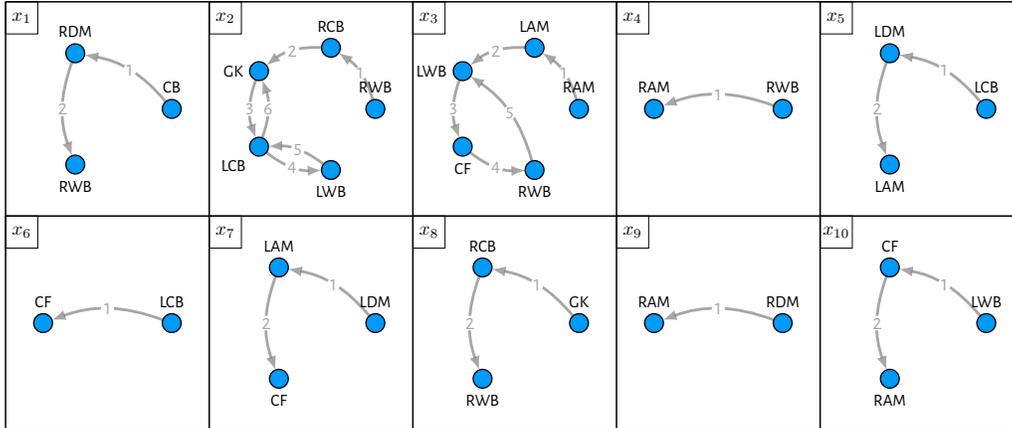


Fig 1: Data from a single match of the StatsBomb data set, where each x_i represents a un-interrupted series of passes by the given team, with indices indicating order of observation, that is, x_i was observed before x_{i+1} . Here vertices correspond to player positions (abbreviated according to Table 1) and edges represent passes, labelled according to the order in which they occurred. These particular observations correspond to the first ten series of passes made by England in their game against Italy during the UEFA Euro 2020 competition.

Another instance of data in this form, and one that will serve as our running example, can be obtained from an in-play football data set shared by StatsBomb.¹ This incredibly rich data set contains high-granularity information concerning events within football matches, and of particular interest to us is information regarding passes. In particular, it is possible from these data to infer, for a given team in a given match, series of un-interrupted passes between their players. Intuitively, each series of passes can be seen as a path over player positions (Figure 1). Moreover, over the course of a football match this will lead to a series of such paths being observed, for example, Figure 1 shows the first ten enacted by England in a match against Italy during the UEFA Euro 2020 competition. As such, a football match (for a given team) can be seen as a sequence or multiset of paths.

Though the problem of eliciting distances between sequences and multisets is not new, little has been done in the way of an overarching review. Moreover, oftentimes these have been addressed separately, with no recognition of the connections inherent from the fact sequences and multisets are closely related. Herein lies the motivation for this work. The intention is for this to serve as a point for reference for anyone faced with data of this structure; which we feel is a very general one. The only restriction we impose is that a distance metric be defined over the underlying space. For each distance, we provide an intuitive interpretation, prove theoretical properties and discuss how they can be computed.

The remainder of this paper will be structured as follows. In Section 2, we introduce the notation to be used throughout and provided background on distance metrics. Section 3 then introduces distances to compare multisets, whilst Section 4 does so for sequences. Finally, in Section 5 we illustrate the use of these distances in practice through an analysis of the StatsBomb data set, where we consider visualising data structure via a dimension reduction technique.

¹<https://github.com/statsbomb/open-data>

2. Background and notation. A single observation will be denoted by X , which may represent either a sequence or multiset. A sequence we denote as follows

$$X = (x_1, \dots, x_N)$$

where $x_i \in \mathcal{X}$ for some general space \mathcal{X} . For example, regarding the football data (Figure 1), \mathcal{X} would denote the space of all paths over the player positions. A multiset is the order-invariant analogue of a sequence and we denote it as follows

$$X = \{x_1, \dots, x_N\}$$

where $x_i \in \mathcal{X}$, with the curly braces $\{\}$ being used to signify this is a multiset and hence the order of elements therein is arbitrary. Note in both we allow $x_i = x_j$ for $i \neq j$, and hence the need to opt for multiset over regular sets. We let $|X| = N$ denote sequence length, or equivalently multiset cardinality.

A multiset X can also be represented via a function $m_X : \mathcal{X} \rightarrow \mathbb{Z}_+$ where $m_X(x)$ denotes the multiplicity of x in X , which we refer to as the *multiplicity function*. Moreover, this defines the support of X in \mathcal{X} as follows

$$\text{Supp}(X) := \{x \in \mathcal{X} : m_X(x) > 0\},$$

denoting the set of unique elements in X . As an example, we might have $\mathcal{X} = \mathbb{Z}_+$ with $X = \{1, 1, 1, 2, 2, 3\}$ a multiset, where $m_X(1) = 3$, $m_X(2) = 2$ and $m_X(3) = 1$, whilst $\text{Supp}(X) = \{1, 2, 3\}$.

A distance measure over the space \mathcal{X} is a function $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_+$, taking as input two elements of the space and outputting some measure of dissimilarity between them. It is natural to require that such functions satisfy certain properties, which are formalised mathematically via the notion of a distance metric.

DEFINITION 2.1 (Distance metric). A function $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_+$ is a *distance metric* over the space \mathcal{X} if, for any $x, y, z \in \mathcal{X}$, the following conditions are satisfied

- (i) $d(x, y) = 0 \iff x = y$ (identity of indiscernibles);
- (ii) $d(x, y) = d(y, x)$ (symmetry);
- (iii) $d(x, y) \leq d(x, z) + d(z, y)$ (triangle inequality);

with the pair (\mathcal{X}, d) being referred to as a *metric space*.

Of particular interest in this work are distance measures between sequences and multisets, so that \mathcal{X} of Definition 2.1 would denote the space of all sequences or multisets over the underlying space \mathcal{X} . As mentioned in the introduction, towards defining such distances it will be assumed that one has access to a distance $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_+$ over the underlying space \mathcal{X} , which we refer to as the *ground distance*. For example, regarding the football data, this would amount to a distance between paths. Moreover, it will be assumed that d satisfies the conditions of Definition 2.1 (with $\mathcal{X} = \mathcal{X}$), and hence is a distance *metric*. In this way, the multiset or sequence X can be seen as collections of points within the metric space (\mathcal{X}, d) .

Finally, we discuss distance normalisation. Often when comparing objects of different sizes via distance measures it can be useful to normalise them. A solution is to use an approach adopted by [Donnat and Holmes \(2018\)](#), based on a metric transformation. This transform is referred to therein as the *Steinhaus transform*, but is also seen in [Deza and Deza \(2009\)](#) where it is referred to as the *biotope transform*. Given a distance metric d (note it must be a metric) over the space \mathcal{X} , with $c \in \mathcal{X}$ some reference element of this space, the Steinhaus transform of d is given by

$$(1) \quad \bar{d}(x, y) := \frac{2d(x, y)}{d(x, c) + d(y, c) + d(x, y)},$$

defining a new distance \bar{d} , which can be shown to be a metric. Note, we leave out any reference to c in this notation, though one should be aware that by definition \bar{d} does depend on it. Observe that since d is a metric, and hence obeys the triangle inequality (iii), we have the following result

$$\begin{aligned} d(x, y) &\leq d(x, c) + d(c, y) \\ \implies 2d(x, y) &\leq d(x, c) + d(y, c) + d(x, y) \\ \implies \bar{d}(x, y) &\leq 1 \end{aligned}$$

that is, \bar{d} is bounded. Moreover, it will be non-negative since it is a ratio of non-negative terms. As such we have $\bar{d}(x, y) \in [0, 1]$ for any $x, y \in \mathcal{X}$.

3. Distances between multisets. In this section, we outline distance measures one can use to compare multisets. All of the distances here share a similar structure, each considering possible relations between elements of either observation, seeking to find the relation which is in some sense ‘optimal’. The objective to optimise is typically some notion of cost, and it is the minimal value of this cost which is taken as the distance. Where these measures differ is in the structure of this relation. For each distance in this section, we give an intuitive and formal definition, discuss theoretical properties and provide details on how they are computed.

3.1. Matching distances. A natural route to defining a distance between multisets is to consider pairing the elements from multiset with the other, defined formally via the notion of a *matching* (Figure 2a). For each matching, one can make use of the ground distance between set elements to define a notion of cost. A distance is then defined by finding the minimum cost matching. The resulting distance also has an alternative interpretation; seen as the minimum cost of turning one multiset into another by (i) inserting or deleting elements with some specified cost or, (ii) substituting one element for another at a cost proportional to their dissimilarity.

Formally, given two multisets X and Y a *matching* is a multiset of pairs

$$(2) \quad \mathcal{M} = \{(x, y) : x \in X, y \in Y\}$$

such that each $x \in X$ is matched to at most one $y \in Y$, taking into account multiplicities, and *vice versa*. Equivalently, each $x \in \text{Supp}(X)$ can be matched to at most $m_X(x)$ elements $y \in \text{Supp}(Y)$, and *vice versa*. Observe by this definition that one must have $0 \leq |\mathcal{M}| \leq \min(|X|, |Y|)$, and a matching which achieves this upper bound is said to be *complete*. For example, the matching of Figure 2a is complete. Finally, we define

$$\mathcal{M}_X := \{x \in X : \exists y \in Y, \text{ with } (x, y) \in \mathcal{M}\}$$

so that $\mathcal{M}_X \subseteq X$ denotes the elements of X which are included in the matching \mathcal{M} , whilst we introduce the shorthand $\mathcal{M}_X^c := X \setminus \mathcal{M}_X$ to denote the elements of X *not* included in the matching \mathcal{M} .

For any given matching \mathcal{M} we can assign it a cost as follows

$$(3) \quad C(\mathcal{M}) = \sum_{(x,y) \in \mathcal{M}} d(x, y) + \lambda(\mathcal{M})$$

with $d(\cdot, \cdot)$ is the ground distance over the underlying space \mathcal{X} and $\lambda(\mathcal{M}) \geq 0$ is some penalty term for un-matched elements, that is, we sum the pairwise distances of matched elements and penalise un-matched elements. A distance between X and Y is now defined by minimising this cost over all matchings.

Each choice for $\lambda(\mathcal{M})$ will define a different distance, and we consider two. For the first, we use the ground distance $d(\cdot, \cdot)$, letting

$$\lambda(\mathcal{M}) = \sum_{x \in \mathcal{M}_x^c} d(x, \Lambda) + \sum_{y \in \mathcal{M}_y^c} d(y, \Lambda)$$

where $\Lambda \in \mathcal{X}$ denotes a reference value, typically taken to be the null or equivalent, with $d(x, \Lambda)$ often capturing some notion of size for the element $x \in \mathcal{X}$, though this will depend on the choice of metric d and the underlying space. For example, if $x \in \mathcal{X}$ are paths we might take Λ to be the empty path.

DEFINITION 3.1 (Matching distance). For two multisets X and Y the matching distance is given by the following

$$(4) \quad d_M(X, Y) := \min_{\mathcal{M}} \left\{ \left(\sum_{(x,y) \in \mathcal{M}} d(x, y) \right) + \sum_{x \in \mathcal{M}_x^c} d(x, \Lambda) + \sum_{y \in \mathcal{M}_y^c} d(y, \Lambda) \right\}$$

where \mathcal{M} denotes a matching of X and Y , and $\Lambda \in \mathcal{X}$ denotes a reference element of \mathcal{X} , typically the null element.

An alternative approach is to penalise each un-matched entry equally by some pre-specified amount $\rho > 0$, that is

$$\begin{aligned} \lambda(\mathcal{M}) &= \rho \times (\# \text{ un-matched elements}) \\ &= \rho(|X| + |Y| - 2|\mathcal{M}|), \end{aligned}$$

which leads to the following distance.

DEFINITION 3.2 (Fixed-penalty matching distance). For two multisets X and Y the fixed-penalty matching distance is given by the following

$$(5) \quad d_{M,\rho}(X, Y) := \min_{\mathcal{M}} \left\{ \sum_{(x,y) \in \mathcal{M}} d(x, y) + \rho(|X| + |Y| - 2|\mathcal{M}|) \right\}$$

where \mathcal{M} is a matching of X and Y , and $\rho > 0$ is a parameter controlling the penalty for un-matched elements.

Computation of these distances requires finding an optimal matching, which can be achieved via the Hungarian algorithm (Kuhn, 1955). This is an algorithm proposed to solve the assignment problem, which seeks an optimal assignment of n ‘workers’ to n ‘tasks’, doing so with a complexity of $\mathcal{O}(n^4)$. As we detail in Appendix A.1, by setting up the right optimisation problem, we can obtain these two distances at a computational complexity of $\mathcal{O}(\max(N, M)^4 + NM)$, where $N = |X|$ and $M = |Y|$, with the $\max(N, M)^4$ term due to the Hungarian algorithm, whilst the NM term arises through the need to evaluate all pairwise distances between elements of X and Y .

We now discuss some theoretical properties of both distances, proofs of which can be found in Appendix B.1. Firstly, both are distance metrics (Definition 2.1).

PROPOSITION 3.3. *Both d_M and $d_{M,\rho}$ satisfy metric conditions (i)-(iii).*

We also have results regarding the form of optimal matchings for each distance, which are particularly useful when it comes to evaluating these distances via the Hungarian algorithm.

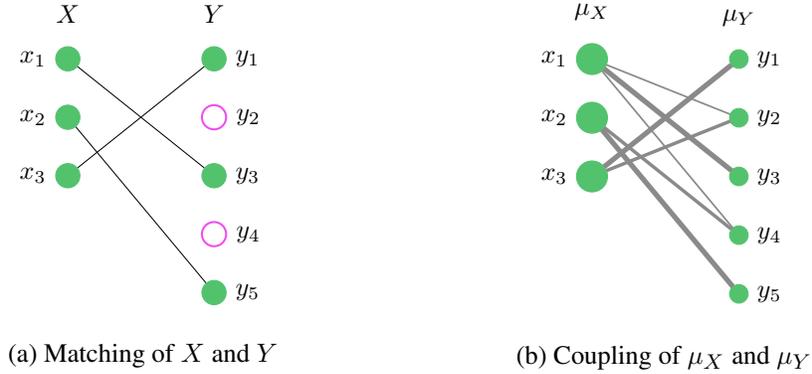


Fig 2: Example relations found when evaluating multiset distances, with (a) showing a matching \mathcal{M} of the multisets $X = \{x_1, \dots, x_3\}$ and $Y = \{y_1, \dots, y_5\}$, whilst (b) shows a coupling \mathbf{P} of the distributions μ_X and μ_Y , where the edge from x_i to y_j is proportional to \mathbf{P}_{ij} , the mass moved from $x_i \in \mathcal{X}$ to $y_j \in \mathcal{X}$, and node (circle) radii of x_i and y_i are proportional to $\mu_X(x_i)$ and $\mu_Y(y_j)$, respectively. For simplicity, here we assume the elements of X and Y are distinct, so that within μ_X and μ_Y the masses are equal.

PROPOSITION 3.4. *For the matching distance d_M , there always exists a complete matching achieving the optimum of eq. (4).*

PROPOSITION 3.5. *For the fixed-penalty matching distance $d_{M,\rho}$ with*

$$\rho \geq \frac{1}{2} \left[\max_{x \in X, y \in Y} d(x, y) \right]$$

there exists a complete matching achieving the optimum of eq. (5).

As consequence of Propositions 3.4 and 3.5, one only needs to optimise over complete matchings when the relevant conditions hold. As such, the size of optimisation problem to be solved via the Hungarian algorithm can be minimised (Appendix A.1).

Proposition 3.5 also sheds light on how one might choose ρ . Following the rationale that when an optimal matching is incomplete one is to some extent ignoring information, it makes sense to choose ρ such that a complete optimal matching can always be found. From Proposition 3.5, one can see this will depend on the pairwise distances of X and Y . However, if the ground distance d happens to be a bounded, that is $d(x, y) \leq K$ for all $x, y \in \mathcal{X}$ and $0 < K < \infty$, then by Proposition 3.5 if $\rho \geq K/2$ one is guaranteed to have a complete optimal matching. Moreover, if one would like as much of the distance to be driven by the pairwise distances as possible then $\rho = K/2$ is a sensible choice. Though having a bounded distance appears restrictive, recall that via the Steinhaus transform of eq. (1) one can obtain a bounded distance with $K = 1$ given *any* distance metric.

We finish by noting that both matching distances are similar to those proposed by others. In particular, Ramon and Bruynooghe (2001) and Eiter and Mannila (1997) both considered the problem of comparing sets within a metric space, though they considered genuine sets whereas we consider multisets, with both defining their respective measures via an optimal relationship between the two sets. Of the two, Ramon and Bruynooghe (2001) is most similar, and in fact the vernacular and notation of matchings that we adopted here was inspired by theirs.

3.2. *Earth mover's distance.* Though theoretically sound, a drawback of the matching distances is that when X and Y are of different sizes the pairwise information of certain elements is to an extent ignored, with the contribution of un-matched elements coming solely via the penalisation terms. Towards defining a distance which avoids such issues, one can use ideas from the literature on Optimal Transport (OT) (Peyré and Cuturi, 2019), an area of research which has considered the problem of quantifying the dissimilarity of probability distributions over general metric spaces. Namely, by converting multisets to distributions an OT-based distance thereof can serve as a proxy for a distance between the original observations.

We convert a multiset X to a distribution as follows. Define $\mu_X : \mathcal{X} \rightarrow [0, 1]$ via

$$(6) \quad \mu_X(x) := \frac{m_X(x)}{|X|},$$

with $\mu_X(x)$ seen as the probability mass located at $x \in \mathcal{X}$. Given two multisets X and Y , we now consider using an OT-based distance between μ_X and μ_Y to measure their dissimilarity; namely the 1-Wasserstein distance (Peyré and Cuturi, 2019, Prop. 2.2), also known as the *earth mover's distance* (EMD).

The EMD admits the following intuition. One imagines that μ_X and μ_Y represent quantities of mass at various locations within the space \mathcal{X} , that is, there is $\mu_X(x)$ mass at $x \in \mathcal{X}$ and $\mu_Y(y)$ mass at $y \in \mathcal{X}$. Moreover, one considers transforming μ_X into μ_Y by ‘transporting’ the mass from one set of locations to the other. Assuming the cost of moving mass between two points is proportional to their distance, that is, moving one unit of mass from $x \in \mathcal{X}$ to $y \in \mathcal{X}$ incurs a cost $d(x, y)$, the EMD is then defined to be the minimum cost required to transform μ_X into μ_Y .

Formally, this can be cast as a linear optimisation problem. Note that by definition, any μ_X and μ_Y have non-zero mass at a finite number of points in the space \mathcal{X} , namely at $\text{Supp}(X) = \{x_1, \dots, x_K\}$ and $\text{Supp}(Y) = \{y_1, \dots, y_L\}$, respectively. As such, in transforming $\mu_X \rightarrow \mu_Y$ we need only consider the movement of mass between this finite collection of locations. The decision variables will now be the mass sent from $x_i \in \text{Supp}(X)$ to $y_j \in \text{Supp}(Y)$ for each pair (x_i, y_j) , which we denote \mathbf{P}_{ij} and collate into the $K \times L$ matrix \mathbf{P} (Figure 2b). Furthermore, with \mathbf{D} the $K \times L$ matrix of pairwise distances, where $\mathbf{D}_{ij} = d(x_i, y_j)$, the goal is to find a \mathbf{P} minimising the total cost, that is

$$\min \sum_{i=1}^K \sum_{j=1}^L \mathbf{D}_{ij} \mathbf{P}_{ij}$$

subject to the constraints

$$\sum_{j=1}^L \mathbf{P}_{ij} = \mu_X(x_i) \quad (\text{for } i = 1, \dots, K) \quad \text{and} \quad \sum_{i=1}^K \mathbf{P}_{ij} = \mu_Y(y_j) \quad (\text{for } j = 1, \dots, L)$$

which ensure that \mathbf{P} defines a movement of mass which starts with the distribution μ_X and ends with μ_Y , as desired. The EMD is subsequently defined to be the total cost of an optimal \mathbf{P} .

Towards a more succinct definition, we adopt notation of Peyré and Cuturi (2019), denoting a set of feasible \mathbf{P} as follows

$$\mathbf{U}(\mu_X, \mu_Y) := \{\mathbf{P} \in \mathbb{R}_+ : \mathbf{P} \mathbf{1}_L = \mathbf{p}_X, \mathbf{P}^T \mathbf{1}_K = \mathbf{p}_Y\}$$

where $\mathbf{1}_N = (1, \dots, 1)$ is the length N vector of ones, and

$$\mathbf{p}_X = (\mu_X(x_1), \dots, \mu_X(x_K)) \quad \mathbf{p}_Y = (\mu_Y(y_1), \dots, \mu_Y(y_L))$$

denote probability vectors associated with each distribution. Adopting the vernacular therein, we refer to any $\mathbf{P} \in \mathbf{U}(\mu_X, \mu_Y)$ as a *coupling* of μ_X and μ_Y . With this, the EMD can be defined as follows.

DEFINITION 3.6 (Earth mover’s distance). For two multisets X and Y the earth movers distance is given by the following

$$(7) \quad d_{\text{EMD}}(X, Y) := \min_{\mathbf{P} \in \mathbf{U}(\mu_X, \mu_Y)} \sum_{i=1}^K \sum_{j=1}^L \mathbf{D}_{ij} \mathbf{P}_{ij}$$

where μ_X and μ_Y are the distributions obtained from X and Y as defined by eq. (6).

Computation of the EMD reduces to solving a linear optimisation problem; specifically the transportation problem. As such, one can appeal to literature on solvers thereof (details can be found in [Peyré and Cuturi, 2019](#), Ch. 3). There also exist packages in various programming languages which can be used to implement these algorithms easily, for example, the Python Optimal Transport (POT) toolbox ([Flamary et al., 2021](#)).

We now consider theoretical properties of d_{EMD} as a distance between multisets. Since the EMD is a distance metric between probability distributions ([Peyré and Cuturi, 2019](#), Prop. 2.2), some properties will be naturally inherited. However, thanks to the normalisation enacted when constructing distributions via eq. (6), not all of the metric conditions will hold, as summarised by the following result (proof in [Appendix B.1](#)).

PROPOSITION 3.7. *The earth mover’s distance d_{EMD} satisfies metric conditions (ii) (symmetry) and (iii) (triangle inequality), but fails (i) (identity of indiscernibles).*

The failure of condition (i) occurs when one multiset is a multiple of the other, that is, if there is some $C > 0$ such that $m_X(x) = C \cdot m_Y(x)$ for all $x \in \mathcal{X}$. However, assuming the multisets X and Y , and the underlying space \mathcal{X} , are all of reasonable size, the chances of this occurring are likely to be low. As such, though [Proposition 3.7](#) may appear unattractive, the practical consequences are unlikely to be severe; though this will clearly depend on how one intends to use the distance. In any case, if necessary, one can extend d_{EMD} to define a valid metric as follows.

DEFINITION 3.8 (Earth mover’s distance with cardinality comparison).

$$(8) \quad d_{\text{sEMD}}(X, Y) := \tau d_{\text{EMD}}(X, Y) + (1 - \tau) d_s(|X|, |Y|)$$

where $d_s(\cdot, \cdot)$ denotes a distance metric between integer values, whilst $0 < \tau < 1$ controls the relative contributions of d_{EMD} and d_s to the overall distance.

PROPOSITION 3.9. *The distance d_{sEMD} satisfies metric conditions (i)-(iii).*

Again, we note this approach to compare multisets via the EMD is not a new idea. For example, [Kusner et al. \(2015\)](#) did exactly this to define a distance between text documents.

4. Distances between sequences. In this section, we turn to the problem of measuring the dissimilarity of sequences, introducing two distances taken from the time series literature. These are typically interpreted as some form of minimum cost transformation, but can also be defined via an optimal relation between the two observations, much like the multiset distances. Again, for each distance we give an intuitive and formal definition before discussing theoretical and computational aspects.

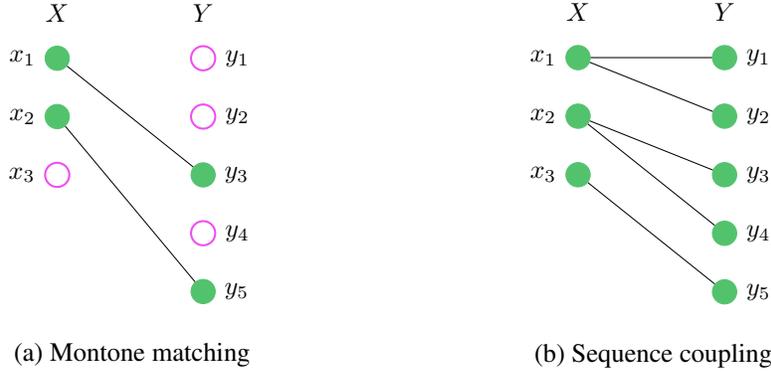


Fig 3: Example relations used to define sequence distances, where (a) shows an example of a monotone matching of the two sequences X and Y , used to define the edit distances, whilst (b) shows a coupling, used to define the dynamic time warping distances.

4.1. *Edit distances.* Here we introduce what we call the *edit distance*, closely related to the Geometric Edit Distance (GED) (Gold and Sharir, 2018; Fox and Li, 2019). As the name suggests, this can be seen as the minimum cost of transforming one sequence into the other by (i) substituting one entry for another at a cost proportional to their dissimilarity, and (ii) inserting and deleting entries with some pre-specified penalty. It is also closely related to the matching distances (Section 3.1), admitting an alternative interpretation via an optimal matching between the two sequences; though the matching must in this case satisfy extra conditions to ensure the preservation of order. It is via this latter interpretation that we provide a formal definition.

Suppose that $X = (x_1, \dots, x_N)$ and $Y = (y_1, \dots, y_M)$ are the two sequences to be compared. Observe the notion of a matching as stated in eq. (2) continues to make sense for sequences. However, since entries have an ordering one can further constrain the form of this matching. Namely, following Gold and Sharir (2018), a matching \mathcal{M} of X and Y is said to be *monotone* if for any $(x_{i_1}, y_{j_1}), (x_{i_2}, y_{j_2}) \in \mathcal{M}$ we have

$$i_1 < i_2 \iff j_1 < j_2$$

which ensures that \mathcal{M} preserves the ordering of each sequence, or more informally and visually, when one draws the matching no lines cross (Figure 3a).

Following the approach taken for the matching distances (Section 3.1), by assigning a cost to each matching, a distance can be defined by minimising the cost over all feasible matchings. Our choice of cost functions for sequences will have similar features as the cost functions for multisets, where we (i) sum pairwise distances of matched entries, and (ii) penalise un-matched entries. Each choice of penalty again defines a different distance, and we consider using the exact same penalties as in Section 3.1, defining the edit distance and fixed-penalty edit distance as follows.

DEFINITION 4.1 (Edit distance). For two sequences X and Y , the edit distance is given by the following

$$(9) \quad d_E(X, Y) := \min_{\mathcal{M}} \left\{ \left(\sum_{(x,y) \in \mathcal{M}} d(x, y) \right) + \sum_{x \in \mathcal{M}_X^c} d(x, \Lambda) + \sum_{y \in \mathcal{M}_Y^c} d(y, \Lambda) \right\}$$

where \mathcal{M} denotes a monotone matching of X and Y , and $\Lambda \in \mathcal{X}$ denotes a reference element of \mathcal{X} , typically the null element.

DEFINITION 4.2 (Fixed-penalty edit distance). For two sequences X and Y the fixed-penalty edit distance is given by the following

$$(10) \quad d_{E,\rho}(X, Y) := \min_{\mathcal{M}} \left\{ \sum_{(x,y) \in \mathcal{M}} d(x, y) + \rho(|X| + |Y| - 2|\mathcal{M}|) \right\}$$

where \mathcal{M} is a monotone matching of X and Y , and $\rho > 0$ is a parameter controlling the penalty for un-matched entries.

Notice Definitions 4.1 and 4.2 are more-or-less identical to Definitions 3.1 and 3.2; the difference being monotonicity of matchings. As with the matching distances, both can be shown to satisfy all metric conditions, summarised via the following result (proofs in Appendix B.2).

PROPOSITION 4.3. Both d_E and $d_{E,\rho}$ satisfies metric conditions (i)-(iii)

Regarding computation, both distances can be evaluated via dynamic programming at a complexity $\mathcal{O}(|X| \cdot |Y|)$, further details of which can be found in Appendix A.2.

4.2. *Dynamic time warping.* Though the edit distances come with the theoretical benefits of being metrics, when faced with observations of differing lengths, much like the matching distances, they only really take into account pairwise information of matched entries, effectively ignoring un-matched ones. This similarly motivates the need for a distance without such a feature. Interestingly, an answer can be found with another distance often seen in the time series literature. Namely, the *dynamic time warping* (DTW) distance (Gold and Sharir, 2018).

In defining the DTW distance, we will use the notation and vernacular of Gold and Sharir (2018). Like the edit distance, the DTW distance is based upon finding a minimum cost relation between the two sequences. The key difference between the two is the form of this relation; where the edit distance considered a monotone matching, the DTW considers a *coupling* of the two sequences (Figure 3b). Note this sequence-based coupling differs from the coupling of distributions used to define the EMD (Section 3.2). Given two sequences X and Y , a coupling is a sequence of pairs $\mathcal{C} = (p_1, \dots, p_R)$, where each $p_r = (x_i, y_j)$ for some with $1 \leq i \leq N$ and $1 \leq j \leq M$. To be a coupling, \mathcal{C} must have the first and last entries paired together, that is $p_1 = (x_1, y_1)$ and $p_R = (x_N, y_M)$, and must satisfy the following

$$p_r = (x_i, y_j) \implies p_{r+1} \in \{(x_i, y_{j+1}), (x_{i+1}, y_j), (x_{i+1}, y_{j+1})\},$$

that is, given x_i and y_j are paired, one can either (i) pair the next two entries x_{i+1} and y_{j+1} , or (ii) enact some *warping*, where an entry from either sequence is paired with more than one from the other. For example, in Figure 3b we see warping for the first and second entries of X . Notice that by definition every entry of one sequence will always be coupled with at least one entry from the other.

To define a distance, one now assigns each coupling \mathcal{C} a cost by summing the pairwise distances of coupled entries before minimising this cost over all couplings, leading to the following.

DEFINITION 4.4. Given sequences X and Y , the dynamic time warping distance is given by the following

$$(11) \quad d_{\text{DTW}}(X, Y) := \min_{\mathcal{C}} \left\{ \sum_{(x,y) \in \mathcal{C}} d(x, y) \right\}$$

where \mathcal{C} is a coupling.

It should be noted the DTW distance has certain theoretical shortcomings. Specifically, it violates the identity of indiscernibles (i) and the triangle inequality (iii). This we summarise with the following result, a proof of which can be found in Appendix B.

PROPOSITION 4.5. *The dynamic time warping distance d_{DTW} satisfies metric condition (ii) (symmetry), but violates conditions (i) (identity of indiscernibles) and (iii) (triangle inequality).*

Depending on the desired application, this may or may not be a significant issue. In the former case, it can be helpful to consider whether one can ensure satisfaction of at least one of these conditions. This motivates the following extension, obtained by inclusion of a warping penalty.

DEFINITION 4.6. Given sequences X and Y , the fixed-penalty DTW distance is given by the following

$$(12) \quad d_{\text{DTW},\rho}(X, Y) := \min_{\mathcal{C}} \left\{ \sum_{(x,y) \in \mathcal{C}} d(x, y) + \rho \cdot w(\mathcal{C}) \right\}$$

where

$$w(\mathcal{C}) := |\{(x_i, y_j) \in \mathcal{C} : (x_i, y_{j+1}) \in \mathcal{C} \text{ or } (x_{i+1}, y_j) \in \mathcal{C}\}|,$$

quantifies the amount of warping in \mathcal{C} , whilst $\rho > 0$ is a parameter controlling the penalisation incurred for each instance of warping.

As a result of introducing this warping penalty the distance now satisfies the identity of indiscernibles (i), as summarised in the following result.

PROPOSITION 4.7. *The fixed penalty dynamic time warping distance $d_{\text{DTW},\rho}$ satisfies metric conditions (i) (identity of indiscernibles) and (ii) (symmetry), but violates (iii) (triangle inequality).*

Regarding computation, both DTW distances can be evaluated via dynamic programming at a time complexity of $\mathcal{O}(|X| \cdot |Y|)$, with further details found in Appendix A.3.

5. Data analysis: Embedding football matches. Returning to the in-play football data, we now show how the distances of Sections 3 and 4 can be used to visualise the structure present therein. In particular, given a choice of distance, we use MDS to obtain a two-dimensional representation of the data, often referred to as an embedding, which can then be plotted.

With the StatsBomb data processed into paths (Figure 1), we are left with a sample

$$X^{(1)}, \dots, X^{(n)}$$

where each $X^{(i)}$ represents all pass sequences enacted by a particular team in a single match. Note each will lead to two sequences or multisets (one for each team), and for this data set we have 1096 games, leading to $n = 2192$ observations. Depending on whether one would like to take order into account, these can be represented as sequences or multisets, that is

$$X^{(i)} = \left(x_1^{(i)}, \dots, x_{N^{(i)}}^{(i)} \right) \quad \text{or} \quad X^{(i)} = \left\{ x_1^{(i)}, \dots, x_{N^{(i)}}^{(i)} \right\}$$

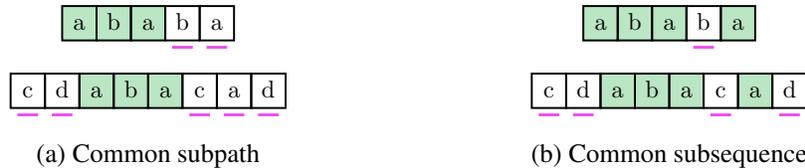


Fig 4: A comparison of common subsequences and subpaths. In (a) and (b) we see the same pair of paths, with (a) highlighting a common subpath, as indicated by shaded (green) entries, whilst (b) shows a common subsequence. In both cases, these are in fact maximal.

where $x_j^{(i)}$ denotes the j th path appearing in the i th observation, as in Figure 1

For a given distance between multisets or sequences, MDS outputs an embedding of data points into m -dimensional Euclidean space such that the pairwise distances between data points are best preserved. More specifically, each data point $X^{(i)}$ gets associated a vector $\mathbf{x}_i \in \mathbb{R}^m$ such that $\|\mathbf{x}_i - \mathbf{x}_j\|_2 \approx d(X^{(i)}, X^{(j)})$ for each pair (i, j) , where $\|\cdot\|_2$ denotes the Euclidean norm. Though the dimension m is general, typically we take $m = 2$ so that each \mathbf{x}_i can be plotted in 2-dimensional space, thus providing a visual summary of the structure present in the observed sample (with respect to the chosen distance).

In this analysis, we compare embeddings obtained in this manner for four different distances, two for sequences and multisets respectively. In particular, we consider the following

- Sequence distances
 1. $\bar{d}_{E,\rho}$: Fixed-penalty edit distance (Definition 4.1), normalised via the Steinhaus transform of eq. (1);
 2. d_{DTW} : Dynamic time warping distance (Definition 4.4);
- Multiset distances
 1. $\bar{d}_{M,\rho}$: Fixed-penalty matching distance (Definition 3.2), normalised via the Steinhaus transform of eq. (1);
 2. d_{EMD} : Earth mover's distance (Definition 3.6).

We must also choose our ground distance, which in this case amounts to specifying a distance metric between paths. A natural approach here is to consider finding maximally-sized common substructures, such as subpaths and subsequences (Figure 4). Suppose $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_m)$ are two paths, where $x_i, y_j \in \mathcal{V}$ for some set of vertices \mathcal{V} , for example, for the football data we have \mathcal{V} denoting the set of player positions. One can now define the longest common subsequence (LCS) distance between x and y as follows

$$d_{LCS}(x, y) := n + m - 2\delta_{LCS}$$

where δ_{LCS} is the maximum length of any subsequence shared by both x and y . Intuitively, this can be seen as the number of entries of either path *not* included in this maximum common subsequence (underlined entries in Figure 4b), or equivalently the minimum number of entries one must delete and insert to transform one path into other. For example, the paths in Figure 4b would have a LCS distance of 5. Further details regarding this distance (and its subpath analogue), including proofs of metric conditions and details regarding computation, can be found in Appendix C.

The data analyst is free to choose the penalisation parameter $\rho > 0$ in $\bar{d}_{E,\rho}$ and $\bar{d}_{M,\rho}$. Here we also consider normalising the ground distance via the Steinhaus transform eq. (1), leading to a ground distance of \bar{d}_{LCS} , so that by the rationale discussed in Section 3.1 we take $\rho = 0.5$ (since $\bar{d}_{LCS}(x, y) \leq 1$).

Figure 5 visualises the embeddings obtained for each of these distances. Note, to simplify the comparison the embeddings have been aligned via rotation and reflection, allowable since

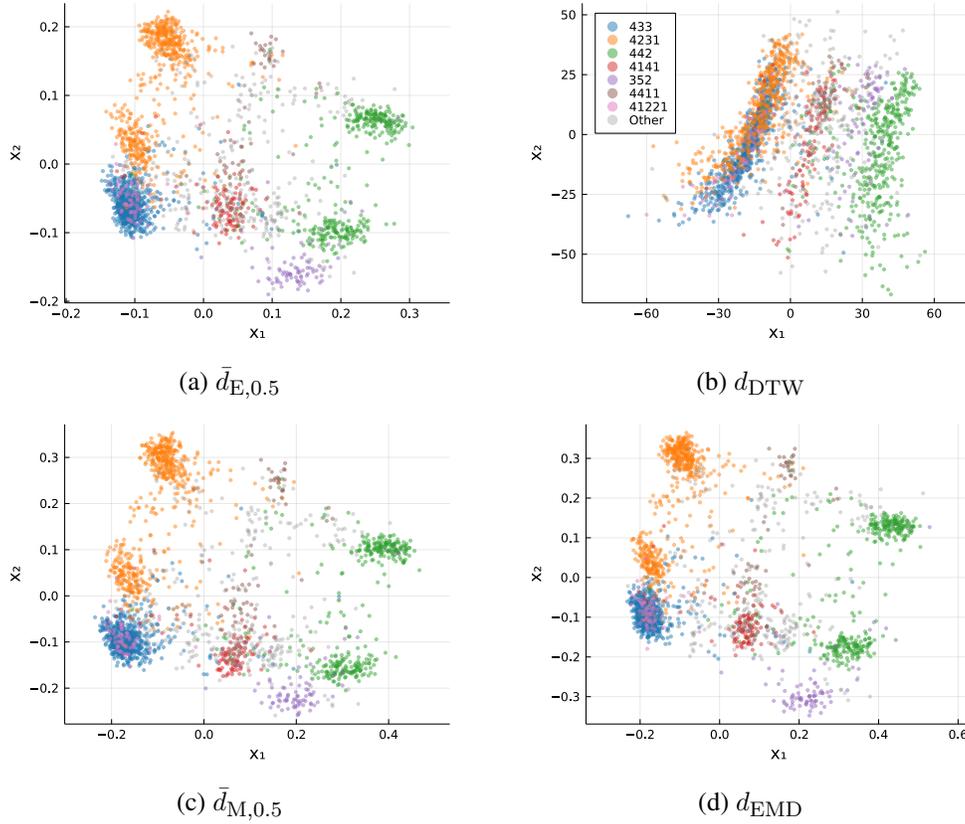


Fig 5: Embeddings of football matches via different distance measures. In each subplot, data points corresponds to a particular team in a given match, labelled to indicate the formation in which a team spent the most time, where for example "433" denotes a formation consisting of 4 defenders, 3 midfielders and 3 attackers. Here (a) and (b) show embeddings obtained via sequence distances, namely the the (normalised) fixed-penalty edit and DTW distances, whilst (c) and (d) show those obtained via two multiset distances, in particular, the (normalised) fixed-penalty matching and EMD distances, respectively.

pairwise Euclidean distances are preserved under such transformations. Here we observe a strong similarity in structure across Figures 5a, 5c and 5d, with each showing clear clusters of data points. In contrast, in Figure 5b we see an embedding which is qualitatively different from the others. To highlight what might be driving the structure observed across these embeddings, data points have also been labelled according to the formation a team was playing most often, so that a data point labelled "433" implies the given team spent most of the time in the corresponding match with a formation consisting of 4 defenders, 3 midfielders and 3 attackers. Here one can observe in Figures 5a to 5d that positions in the respective embedded space appear to be congruent with the formation a team was playing, so that two data points which are near one another therein are likely to be using a similar formation. Moreover, for Figures 5a, 5c and 5d there is a strong correspondence between clusters and formations, with the "433" formation being a good example, though some formations, such as "4231" and "442", appear to have more than one cluster.

6. Discussion. In this paper, we have considered the problem of measuring the dissimilarity of sequences and multisets. Drawing on the wider literature, we have discussed various

distances one can invoke, all of which make use of a pre-specified ground distance over the underlying space. For each distance, we have given a high-level intuition, proved theoretical properties and outlined how they can be computed. For certain distances, such as the EMD and DTW distance, we also propose extensions which allow the distances to satisfy additional metric conditions. Finally, we have illustrated how these distances can be used in practice through a novel analysis of an in-play football data set shared by StatsBomb, where we are able to uncover the squad formation based purely on passes between players.

Regarding future work, one could firstly consider whether other distances could be (or have been) defined. For example, can we consider an analogue of the EMD distance for sequences? One could also study through simulation what features each distance can take into account, providing guidance on which distance to use for a given problem or question of interest. There is also scope to expand on the data analysis of this work. For example, one could consider measuring quantitatively the relationship between formation and distance in the StatsBomb data, similar to the analysis of [Donnat and Holmes \(2018\)](#). Finally, note that often one can first aggregate observations to some other form before measuring their distance. As such, it is natural to ask whether one gains anything by using the distances discussed here? For example, the football data could be collapsed to a vector of counts over player positions, or perhaps a multigraph, encoding the number of passes observed between each pair of positions. Distances between these aggregates are likely to be much faster to compute than those between a sequence or multiset of paths, however, there is going to be a loss of information incurred through aggregation. It would be interesting to explore, either through a simulation study or real-data analysis, whether one gains something by taking into account this extra information via use of a multiset, or sequence, distance instead of an aggregate-based distance.

REFERENCES

- BECHT, E., MCINNES, L., HEALY, J., DUTERTRE, C.-A., KWOK, I. W., NG, L. G., GINHOUX, F. and NEWELL, E. W. (2019). Dimensionality reduction for visualizing single-cell data using UMAP. *Nature biotechnology* **37** 38–44.
- DEZA, M. M. and DEZA, E. (2009). *Encyclopedia of Distances*. Springer. <https://doi.org/10.1007/978-3-642-00234-2>
- DONNAT, C. and HOLMES, S. (2018). Tracking network dynamics: A survey using graph distances. *Annals of Applied Statistics* **12** 971–1012. <https://doi.org/10.1214/18-AOAS1176>
- EITER, T. and MANNILA, H. (1997). Distance measures for point sets and their computation. *Acta informatica* **34** 109–133.
- ESTER, M., KRIEGEL, H.-P., SANDER, J., XU, X. et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd* **96** 226–231.
- FLAMARY, R., COURTY, N., GRAMFORT, A., ALAYA, M. Z., BOISBUNON, A., CHAMBON, S., CHAPEL, L., CORENFLOS, A., FATRAS, K., FOURNIER, N., GAUTHERON, L., GAYRAUD, N. T. H., JANATI, H., RAKOTOMAMONJY, A., REDKO, I., ROLET, A., SCHUTZ, A., SEGUY, V., SUTHERLAND, D. J., TAVENARD, R., TONG, A. and VAYER, T. (2021). POT: Python optimal transport. *Journal of Machine Learning Research* **22** 1–8.
- FOX, K. and LI, X. (2019). Approximating the geometric edit distance. *Leibniz International Proceedings in Informatics, LIPIcs* **149**. <https://doi.org/10.4230/LIPIcs.ISAAC.2019.23>
- GOLD, O. and SHARIR, M. (2018). Dynamic time warping and geometric edit distance: breaking the quadratic barrier. *ACM Transactions on Algorithms* **14**. <https://doi.org/10.1145/3230734>
- HASTIE, T., TIBSHIRANI, R., FRIEDMAN, J. H. and FRIEDMAN, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* **2**. Springer.
- IZENMAN, A. J. (2008). Modern multivariate statistical techniques. *Regression, classification and manifold learning* **10** 978–0.
- KUHN, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* **2** 83–97. <https://doi.org/10.1002/nav.3800020109>
- KUMAR, R. and VASSILVITSKII, S. (2010). Generalized distances between rankings. In *Proceedings of the 19th international conference on World wide web* 571–580.

- KUSNER, M., SUN, Y., KOLKIN, N. and WEINBERGER, K. (2015). From word embeddings to document distances. In *International conference on machine learning* 957–966. PMLR.
- MCINNIS, L., HEALY, J. and MELVILLE, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.
- PEYRÉ, G. and CUTURI, M. (2019). Computational optimal transport. *Foundations and Trends in Machine Learning* **11** 1-257. <https://doi.org/10.1561/22000000073>
- RAEDER, T. and CHAWLA, N. V. (2011). Market basket analysis with networks. *Social network analysis and mining* **1** 97–113.
- RAMON, J. and BRUYNOOGHE, M. (2001). A polynomial time computable metric between point sets. *Acta Informatica* **37** 765-780. <https://doi.org/10.1007/PL00013304>
- WAGNER, R. A. and FISCHER, M. J. (1974). The string-to-string correction problem. *Journal of the ACM (JACM)* **21** 168-173. <https://doi.org/10.1145/321796.321811>
- YANG, D., ZHANG, D., ZHENG, V. W. and YU, Z. (2015). Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **45** 129-142. <https://doi.org/10.1109/TSMC.2014.2327053>

Position	Abbreviation
Center Back	CB
Right Defensive Midfield	RDM
Right Wing Back	RWB
Right Center Back	RCB
Goal Keeper	GK
Left Center Back	LCB
Left Wing Back	LWB
Right Attacking Midfield	RAM
Left Attacking Midfield	LAM
Center Forward	CF
Left Defensive Midfield	LDM
Right Back	RB
Right Wing	RW
Left Wing	LW
Center Defensive Midfield	CDM
Right Center Midfield	RCM
Left Back	LB
Left Center Midfield	LCM
Center Attacking Midfield	CAM

TABLE 1
Abbreviations for player positions used in Figure 1.

APPENDIX A: DISTANCE COMPUTATION

A.1. Matching distances. As mentioned in Section 3.1, we consider evaluating both matching distances via the Hungarian algorithm (Kuhn, 1955), a specialised algorithm proposed to solve the so-called assignment problem. Suppose that one has two sets $A = \{a_1, \dots, a_n\}$ and $B = \{b_1, \dots, b_n\}$, both of size n , then assignment problem considers pairing elements of set A with those of set B in an ‘optimal’ way, where the objective is defined by assigning a cost to each possible pairing. Note the labelling of elements here is arbitrary but will serve a purpose in what follows, allowing us to index set elements.

A pairing of set elements can be encoded via a permutation $\sigma \in S_n$, where S_n denotes the set of all permutation on n symbols, with $\sigma(i) = j$ implying that $a_i \in A$ has been paired with $b_j \in B$. We summarise the cost of pairings in the $n \times n$ matrix C , where C_{ij} denotes the cost incurred when $a_i \in A$ is paired with $b_j \in B$. Now, the assignment problem can be stated formally as finding the minimum cost permutation σ , that is

$$\min_{\sigma \in S_n} \sum_{i=1}^n C_{i, \sigma(i)}$$

the solution of which may not be unique. Observe that though A and B are often assumed to be sets, this formulation works equally well if they are multisets, as we will assume them to be.

Given any square $n \times n$ matrix C , the Hungarian algorithm will return a permutation σ which minimises the cost above. Towards evaluating d_M and $d_{M, \rho}$, we consider constructing a matrix C such that the optimal solution found via the Hungarian algorithm coincides with that required in their respective definitions. Note that due to Propositions 3.4 and 3.5, there are situations in which we need only optimise over complete matchings. In these situations, we can minimise the size of optimisation problem to be solved via the Hungarian algorithm, or equivalently, minimise the size of C , as we outline in Appendix A.1.1. Alternatively, if one would like to optimise over all matchings, a slightly bigger C can be specified, as detailed in Appendix A.1.2.

Given two multisets X and Y the choice of which approach to take for each matching distance can be summarised as follows

- For $d_M(X, Y)$, optimise over complete matchings (Appendix A.1.1)
- For $d_{M,\rho}(X, Y)$, the approach depends on whether ρ satisfies the condition of Proposition 3.5. In particular, letting $K = \max_{x \in X, y \in Y} d(x, y)$ we have
 - If $\rho \geq K/2$, optimise over complete matchings (Appendix A.1.1)
 - If $\rho < K/2$, optimise over all matchings (Appendix A.1.2).

A.1.1. *Optimising over complete matchings.* For multisets X and Y , suppose without loss of generality we have $|X| \leq |Y|$. We construct the $|Y| \times |Y|$ matrix C as follows

$$(13) \quad C_{ij} = \begin{cases} d(x_i, y_j) & \text{if } i \leq |X| \\ \lambda(y_i) & \text{if } i > |X| \end{cases}$$

where $\lambda(y)$ denotes the penalty incurred when $y \in Y$ is not included in the matching, where for the matching distance d_M we let $\lambda(y) = d(y, \Lambda)$, whilst for the fixed-penalty matching distance $d_{M,\rho}$ we let $\lambda(y) = \rho$. Here, to account for the fact that X and Y may be of different sizes, we effectively introduce $|Y| - |X|$ dummy elements which entries from Y can be paired with, where being paired with a dummy element is equivalent to being un-matched. Now, each $\sigma \in S_{|Y|}$ encodes a matching \mathcal{M} of X and Y given by the following

$$\mathcal{M} = \{(x_i, y_{\sigma(i)}) : 1 \leq i \leq |X|\},$$

which includes all elements of X and is thus complete. Moreover, according to C this has the following cost

$$\begin{aligned} C(\mathcal{M}) &= \sum_{i=1}^{|Y|} C_{i, \sigma(i)} \\ &= \sum_{(x,y) \in \mathcal{M}} d(x, y) + \sum_{y \in \mathcal{M}_Y^c} \lambda(y) \\ &= \sum_{(x,y) \in \mathcal{M}} d(x, y) + \sum_{x \in \mathcal{M}_X^c} \lambda(x) + \sum_{y \in \mathcal{M}_Y^c} \lambda(y) \end{aligned}$$

where the last line follows since $\mathcal{M}_X^c = \emptyset$ by virtue of \mathcal{M} being complete. As such, in optimising over the permutations σ via the Hungarian algorithm we are effectively enacting the following optimisation

$$\min_{\mathcal{M}} \left\{ \sum_{(x,y) \in \mathcal{M}} d(x, y) + \sum_{x \in \mathcal{M}_X^c} \lambda(x) + \sum_{y \in \mathcal{M}_Y^c} \lambda(y) \right\}$$

where \mathcal{M} is a *complete* matching. Observe that, with the respective $\lambda(\cdot)$ substituted in, this is almost identical to optimisation of Definitions 3.1 and 3.2, the only difference being that this optimises over only complete matchings.

A.1.2. *Optimising over all matchings.* For multisets X and Y , in this case we construct the $(|X| + |Y|) \times (|X| + |Y|)$ matrix C as follows

$$C_{ij} = \begin{cases} d(x_i, y_j) & \text{if } i \leq |X| \text{ and } j \leq |Y| \\ \lambda(y_i) & \text{if } i > |X| \text{ and } j \leq |Y| \\ \lambda(x_i) & \text{if } i \leq |X| \text{ and } j > |Y| \\ 0 & \text{if } i > |X| \text{ and } j > |Y| \end{cases}$$

where again $\lambda(\cdot)$ denotes the penalisation for un-matched entries as stated in Appendix A.1.1. In this case, we introduce dummy elements to *both* sets, namely $|Y|$ in X and $|X|$ in Y , implying now elements from either set can be un-matched by being paired with a dummy element. Each $\sigma \in S_{|X|+|Y|}$ encodes a matching \mathcal{M} of X and Y given by the following

$$\mathcal{M} = \{(x_i, y_{\sigma(i)}) : 1 \leq i \leq |X|, \sigma(i) \leq |Y|\},$$

where we must include the extra constraint $\sigma(i) \leq |Y|$ since in this case elements of X can be paired with dummy elements, that is, be un-matched. By the definition of C , this implies the following cost

$$\begin{aligned} C(\mathcal{M}) &= \sum_{i=1}^{|X|+|Y|} C_{i,\sigma(i)} \\ &= \sum_{(x,y) \in \mathcal{M}} d(x,y) + \sum_{x \in \mathcal{M}_x^c} \lambda(x) + \sum_{y \in \mathcal{M}_y^c} \lambda(y) \end{aligned}$$

where \mathcal{M} is a matching (not necessarily complete). Again, a comparison with Definitions 3.1 and 3.2 reveals the similarity between the minimisation problems therein and that of the assignment problem parameterised by C . Moreover, in this case since we are optimising over *all* matching they are indeed equivalent.

A.2. Edit distances. Both edit distances are special cases of the so-called string edit distance of Wagner and Fischer (1974), and as such the dynamic programming algorithm proposed therein can be invoked to compute them. Consider first the edit distance (Definition 4.1). Supposing that X and Y are the sequences to be compared, introducing the notation $X_{k:l} = (x_k, \dots, x_l)$, the approach is to evaluate $d_E(X_{1:i}, Y_{1:j})$ incrementally until $i = |X|$ and $j = |Y|$ using the following recursive result

$$d_E(X_{1:i}, Y_{1:j}) = \min \begin{cases} d_E(X_{1:(i-1)}, Y_{1:j}) + d(x_i, \Lambda) \\ d_E(X_{1:i}, Y_{1:(j-1)}) + d(y_j, \Lambda) \\ d_E(X_{1:(i-1)}, Y_{1:(j-1)}) + d(x_i, y_j) \end{cases}$$

where here we are essentially comparing three possibilities (i) the i th entry of X is un-matched, (ii) the j th entry of Y is un-matched, and (iii) the i th entry of X is matched with the j th entry of Y . Introducing the notation $C_{ij} = d_E(X_{1:(i-1)}, Y_{1:(j-1)})$ this is equivalent to filling up the matrix C either row-by-row or column-by-column via the following recursive formula

$$C_{(i+1)(j+1)} = \min \begin{cases} C_{i(j+1)} + d(x_i, \Lambda) \\ C_{(i+1)j} + d(y_j, \Lambda) \\ C_{ij} + d(x_i, y_j) \end{cases}$$

where the final entry corresponds to the desired distance, that is $d_E(X, Y) = C_{(|X|+1)(|Y|+1)}$.

Note we add one to all indices here since the first column and row of C function as boundary conditions. These correspond to when $i = 1$ or $j = 1$, that is, when we have values such as $X_{1:0}$ or $Y_{1:0}$ appearing in the recursive definition. Towards specifying these values, we see $X_{1:0}$ as an empty sequence, so that when comparing $X_{1:0}$ to $Y_{1:j}$ each entry of the latter will be un-matched and hence penalised. This implies

$$C_{1(j+1)} = d_E(X_{1:0}, Y_{1:j}) = \sum_{k=1}^j d(y_k, \Lambda),$$

for $j = 1, \dots, |Y|$, whilst by equivalent reasoning we have

$$C_{(i+1)1} = d_E(X_{1:i}, Y_{1:0}) = \sum_{k=1}^i d(x_k, \Lambda),$$

for $i = 1, \dots, |X|$. Finally, we let

$$C_{11} = d_E(X_{1:0}, Y_{1:0}) = 0$$

since $X_{1:0} = Y_{1:0}$ by virtue of both being the empty sequence.

Algorithm 1 outlines pseudocode for the resulting algorithm to evaluate d_E , using this matrix notation. Furthermore, observe that when updating a row (or column) of C one only needs to know the previous row (or column). As such, one need only store the current and previous row, leading to an algorithm which uses less memory and is typically faster. Pseudocode of this light-memory alternative can also be seen in Algorithm 2.

Turning now to the fixed-penalty edit distance (Definition 4.2), the approach more-or-less the same, up to a slight change of the recursive formula. In particular, in this case we have

$$d_{E,\rho}(X_{1:i}, Y_{1:j}) = \min \begin{cases} d_{E,\rho}(X_{1:(i-1)}, Y_{1:j}) + \rho \\ d_{E,\rho}(X_{1:i}, Y_{1:(j-1)}) + \rho \\ d_{E,\rho}(X_{1:(i-1)}, Y_{1:(j-1)}) + d(x_i, y_j) \end{cases}$$

which leads to an analogous definition of matrix C , with its corresponding recursive formula given by

$$C_{(i+1)(j+1)} = \min \begin{cases} C_{i(j+1)} + \rho \\ C_{(i+1)j} + \rho \\ C_{ij} + d(x_i, y_j) \end{cases}$$

with $C_{(|X|+1)(|Y|+1)}$ again corresponding to the desired distance. Moreover, in this case we have

$$\begin{aligned} C_{(i+1)1} &= i\rho \quad (\text{for } i = 0, \dots, |X|) \\ C_{1(j+1)} &= j\rho \quad (\text{for } j = 0, \dots, |Y|). \end{aligned}$$

Pseudocode of the resulting of the resulting algorithm to evaluate $d_{E,\rho}(X, Y)$ can be seen in Algorithm 3, with the light-memory analogue outlined in Algorithm 4.

A.3. Dynamic time warping distances. Similar to the edit distances, the DTW distances can be evaluated via dynamic programming. In fact, the algorithms are almost identical, differing only in the recursive formulae used.

First, we outline how to compute $d_{\text{DTW}}(X, Y)$ for given sequences X and Y , following the implementation of Gold and Sharir (2018), Sec. 3. Using the notation $X_{k:l} = (x_k, \dots, x_l)$, one evaluates $d_{\text{DTW}}(X_{1:i}, Y_{1:j})$ incrementally until $i = |X|$ and $j = |Y|$ via the following recursive result

$$d_{\text{DTW}}(X_{1:i}, Y_{1:j}) = d(x_i, y_j) + \min \begin{cases} d_{\text{DTW}}(X_{1:(i-1)}, Y_{1:j}) \\ d_{\text{DTW}}(X_{1:i}, Y_{1:(j-1)}) \\ d_{\text{DTW}}(X_{1:(i-1)}, Y_{1:(j-1)}) \end{cases}$$

where here one is essentially comparing three possibilities (i) warping on the j th entry of Y , that is, y_j being paired with more than one element of X , (ii) warping on the i th entry of X , and (iii) no warping, with x_i and y_j being paired *only* with each other. Note the $d(x_i, y_j)$ term comes out front of the minimisation since by definition x_i and y_j must be paired.

Introducing the notation $C_{ij} = d_{\text{DTW}}(X_{1:(i-1)}, Y_{1:(j-1)})$, the incremental computation can be seen as filling-up the matrix C either row-by-row or column-by-column via the following recursive formula

$$C_{(i+1)(j+1)} = d(x_i, y_j) + \min \begin{cases} C_{i(j+1)} \\ C_{(i+1)j} \\ C_{ij} \end{cases}$$

with $d_{\text{DTW}}(X, Y) = C_{(|X|+1)(|Y|+1)}$. As with evaluating the edit distances (Appendix A.2), we must also pre-specify the first row and column on C . Here we again assume $C_{11} = 0$ whilst

$$C_{(i+1)1} = \infty \quad (\text{for } i = 1, \dots, |X|) \quad C_{1(j+1)} = \infty \quad (\text{for } j = 1, \dots, |Y|).$$

To see why this is the case, consider the second column entries, that is

$$C_{i2} = d_{\text{DTW}}(X_{1:(i-1)}, Y_{1:1}).$$

Observe that since $Y_{1:1} = (y_1)$ is a sequence with a single entry, the only valid coupling here is where y_1 is paired with every entry of $X_{1:(i-1)}$. By opting for this choice of boundary values for C one essentially ensures this occurs via the recursive formula. In particular, if one considers filling the second column of C , one has

$$C_{22} = d(x_1, y_1) + \min \begin{cases} \infty \\ \infty \\ 0 \end{cases}$$

so we choose the third option, pairing the first two entries with no warping, whilst for $i > 2$ we have

$$C_{i2} = d(x_1, y_1) + \min \begin{cases} C_{(i-1)2} \\ \infty \\ \infty \end{cases}$$

where here we choose the first option, which corresponds to warping on the 1st entry of X , that is, y_1 being paired with more than one element of X . The same reasoning can be used to justify the initial values of the first row by considering filling the second row of C , wherein essentially the roles of X and Y are swapped.

Algorithm 5 outlines the algorithm which fills the matrix C via this recursive formula to obtain the desired distance. As with the edit distances, this procedure only requires knowledge of the previous and current row, and hence a lighter memory alternative can be considered, as detailed in Algorithm 6.

Turning now to the fixed-penalty DTW distance $d_{\text{DTW},\rho}$, the approach is almost identical, albeit with a slight change in the recursive formula. Namely, to evaluate $d_{\text{DTW},\rho}(X_{1:i}, Y_{1:j})$ we use the following

$$d_{\text{DTW},\rho}(X_{1:i}, Y_{1:j}) = d(x_i, y_j) + \min \begin{cases} d_{\text{DTW},\rho}(X_{1:(i-1)}, Y_{1:j}) + \rho \\ d_{\text{DTW},\rho}(X_{1:i}, Y_{1:(j-1)}) + \rho \\ d_{\text{DTW},\rho}(X_{1:(i-1)}, Y_{1:(j-1)}) \end{cases}$$

where we have simply included the ρ term in the cases corresponding to warping. This leads to analogous algorithms to compute $d_{\text{DTW},\rho}(X, Y)$, namely Algorithm 7, which does so by incrementally filling a matrix, and the light-memory approach of Algorithm 8, which stores only the current and previous row.

APPENDIX B: PROOFS

B.1. Multiset distances.

PROOF OF PROPOSITION 3.3 (PART 1). To aide this exposition, we write d_M in terms of its associate cost function as follows

$$d_M(X, Y) = \min_{\mathcal{M}} C(\mathcal{M})$$

where

$$C(\mathcal{M}) = \left(\sum_{(x,y) \in \mathcal{M}} d(x, y) \right) + \sum_{x \in \mathcal{M}_X^c} d(x, \Lambda) + \sum_{y \in \mathcal{M}_Y^c} d(y, \Lambda),$$

denoting the cost of the matching \mathcal{M} .

We first show (i) holds. Assuming $X = Y$, then one can construct a matching \mathcal{M}_1 by pairing equivalent elements of X and Y , leading to the following upper bound on the distance

$$\begin{aligned} d_M(X, Y) &\leq C(\mathcal{M}_1) \\ (14) \quad &= \sum_{(x,y) \in \mathcal{M}_1} d(x, y) + 0 + 0 \\ &= 0 \end{aligned}$$

where the second line follow since \mathcal{M}_1 includes all elements of X and Y and thus no penalisation will occur, whilst the final line follows since \mathcal{M}_1 matches equivalent elements and hence (using the fact $d(\cdot, \cdot)$ is a metric) all pairwise distances will be zero. Note also, since d_M will be a sum of positive values (since $d(\cdot, \cdot)$ is a metric), we also have $d_M \geq 0$. Together this implies $d_M(X, Y) = 0$.

Conversely, assume that $d_M(X, Y) = 0$. This implies that both the matching cost and penalisation terms must be zero. Since we implicitly assume no element is equal to the null element Λ , then the penalty term being zero implies that all elements of X and Y must be included in the matching. Therefore, we have a complete matching with zero cost. Specifically, supposing \mathcal{M}^* is the optimal matching, we have

$$\begin{aligned} d_M(X, Y) &= \sum_{(x,y) \in \mathcal{M}^*} d(x, y) \\ &= 0. \end{aligned}$$

Since d is a metric it is non-negative, and hence

$$d(x, y) = 0, \quad \forall (x, y) \in \mathcal{M}^*,$$

which, again using the fact d is a metric, implies

$$x = y, \quad \forall (x, y) \in \mathcal{M}^*$$

and hence $X = Y$, confirming satisfaction of (i).

The condition (ii) follows trivially from the symmetry of $d(\cdot, \cdot)$ and the penalisation term.

The final condition to show is the triangle inequality (iii). Assuming that $X = \{x_1, \dots, x_n\}$, $Y = \{y_1, \dots, y_m\}$ and $Z = \{z_1, \dots, z_k\}$ are three multisets, we seek to show that

$$d_M(X, Y) \leq d_M(X, Z) + d_M(Z, Y).$$

Now, let \mathcal{M}_1^* and \mathcal{M}_2^* denote optimal matchings for $d_M(X, Z)$ and $d_M(Z, Y)$ respectively, so that

$$d_M(X, Z) = C(\mathcal{M}_1^*) \quad d_M(Z, Y) = C(\mathcal{M}_2^*)$$

writing these as

$$\mathcal{M}_1^* = \{(x_{i_1}, z_{j_1}), \dots, (x_{i_r}, z_{j_r})\} \quad \mathcal{M}_2^* = \{(z_{l_1}, y_{k_1}), \dots, (z_{l_s}, y_{k_s})\}.$$

Now, \mathcal{M}_1^* and \mathcal{M}_2^* induce a matching \mathcal{M}_3 of X and Y as follows

$$(15) \quad \mathcal{M}_3 = \{(x_i, y_j) : (x_i, z_k) \in \mathcal{M}_1^* \text{ and } (z_k, y_j) \in \mathcal{M}_2^* \text{ for some } z_k \in Z\}$$

that is, we pair elements of X and Y if they were paired to the same elements of Z . Notice by definition we have

$$d_M(X, Y) \leq C(\mathcal{M}_3),$$

consequently the triangle inequality will follow if we can show the following holds

$$(16) \quad C(\mathcal{M}_3) \leq d_M(X, Z) + d_M(Z, Y).$$

To prove eq. (16) we consider every possible term on the LHS and show that this is less than or equal to some unique terms appearing on the RHS. The keys terms appearing on the LHS are (i) pairwise distances for matched elements (ii) penalisations of unmatched elements.

We first consider (i). By definition of \mathcal{M}_3 , each pair $(x_i, y_j) \in \mathcal{M}_3$ is associated with some *unique* $(x_i, z_k) \in \mathcal{M}_1^*$ and $(z_k, y_j) \in \mathcal{M}_2^*$, that is, there is some element $z_k \in Z$ which both x_i and y_j are matched to. Furthermore, since $d(\cdot, \cdot)$ is a distance metric it satisfies the triangle inequality, and so

$$d(x_i, y_j) \leq d(x_i, z_k) + d(z_k, y_j),$$

and thus each pairwise distance of matched elements on the LHS of eq. (16) is less than or equal to some unique terms on the RHS.

For (ii) consider first the penalisation terms for elements of X not included in the matching \mathcal{M}_3 , that is $d(x, \Lambda)$ for $x \in (\mathcal{M}_3)_X^c$. We now seek to show that $d(x, \Lambda)$ is less than or equal to some (unique) terms appearing on the RHS of eq. (16). For x to not be in \mathcal{M}_3 one of two things must have happened

1. $(x, z) \in \mathcal{M}_1^*$ for some $z \in Z$ with $(z, y) \notin \mathcal{M}_2^*$ for any $y \in Y$

$$\implies \text{a term on the RHS of } d(x, z) + d(z, \Lambda)$$

which will also be unique to the pair (x, z) . Now, since $d(\cdot, \cdot)$ is a metric it obeys the triangle inequality, thus

$$d(x, \Lambda) \leq d(x, z) + d(z, \Lambda)$$

as desired;

2. Alternatively, we might have $(x, z) \notin \mathcal{M}_1^*$ for any $z \in Z$

$$\implies \text{a term on the RHS of } d(x, \Lambda),$$

and thus in this case we trivially have

$$d(x, \Lambda) \leq d(x, \Lambda).$$

In either case, we have a term on the LHS of eq. (16) which is less than or equal to some unique terms on the RHS. This argument can be applied similarly to the penalisation terms for elements of Y not in the matching \mathcal{M}_3 .

Thus we have that every term on the LHS of eq. (16) is less than or equal to some unique terms on the RHS, proving the inequality holds. Thus d_M satisfies condition (iii), completing the proof. \square

PROOF OF PROPOSITION 3.3 (PART 2). To aide this exposition, we write $d_{M,\rho}$ in terms of its associate cost function as follows

$$d_{M,\rho}(X, Y) = \min_{\mathcal{M}} C(\mathcal{M})$$

where

$$C(\mathcal{M}) = \sum_{(x,y) \in \mathcal{M}} d(x, y) + \rho(|X| + |Y| - 2|\mathcal{M}|),$$

denoting the cost of the matching \mathcal{M} .

We first consider condition (i). Note firstly that since $d(x, y) \geq 0$ (as it is a metric) and $\rho > 0$, this implies $d_{M,\rho}(X, Y) \geq 0$ for all multisets X and Y . Now, assuming $X = Y$, one can construct a matching \mathcal{M}_1 by pairing equivalent elements of X and Y . The existence of this matching leads to the following upper bound on the distance

$$\begin{aligned} d_{M,\rho}(X, Y) &\leq C(\mathcal{M}_1) \\ &= \sum_{(x,y) \in \mathcal{M}_1} d(x, y) \\ &= 0 \end{aligned}$$

where the second line follow since \mathcal{M}_1 included all elements of X and Y and thus no penalisation will occur, whilst the final line follows since \mathcal{M}_1 matches equivalent elements and hence (using the fact that $d(\cdot, \cdot)$ is a metric) all pairwise distances will be zero. This combined with $d_{M,\rho}(X, Y) \geq 0$ implies $d_{M,\rho}(X, Y) = 0$.

Conversely, $d_{M,\rho}(X, Y) = 0$ implies both the sum of pairwise distances and penalisation terms must be zero. Thus, if \mathcal{M}^* is the optimal matching, then all elements of X and Y are included in \mathcal{M}^* and we have

$$d_{M,\rho} = \sum_{(x,y) \in \mathcal{M}^*} d(x, y) = 0.$$

Now, since $d(\cdot, \cdot) \geq 0$ this implies

$$d(x, y) = 0, \quad \forall (x, y) \in \mathcal{M}^*,$$

and hence

$$x = y, \quad \forall (x, y) \in \mathcal{M}^*.$$

Since all elements of either set are included in \mathcal{M}^* this implies $X = Y$, thus confirming satisfaction of (i).

As with Proposition 3.3 (Part 1), the symmetry condition (ii) follows trivially from the symmetry of $d(x, y)$ and the penalisation term.

Finally, we verify condition (iii), the triangle inequality. Here we follow exactly the same steps seen in the proof of Proposition 3.3 (Part 1). Assuming that $X = \{x_1, \dots, x_n\}$, $Y = \{y_1, \dots, y_m\}$ and $Z = \{z_1, \dots, z_k\}$ are three multisets, we seek to show that

$$d_{M,\rho}(X, Y) \leq d_{M,\rho}(X, Z) + d_{M,\rho}(Z, Y).$$

Now, let \mathcal{M}_1^* and \mathcal{M}_2^* denote optimal matchings for $d_{M,\rho}(X, Z)$ and $d_{M,\rho}(Z, Y)$ respectively, that is

$$d_{M,\rho}(X, Z) = C(\mathcal{M}_1^*) \quad d_{M,\rho}(Z, Y) = C(\mathcal{M}_2^*)$$

then these again induce a matching \mathcal{M}_3 of X and Y as in eq. (15). Moreover, following the reasoning therein, the triangle inequality will follow if we can show the following inequality holds

$$(17) \quad C(\mathcal{M}_3) \leq d_{M,\rho}(X, Z) + d_{M,\rho}(Z, Y),$$

where the only difference here is in the form of cost function.

As in the proof of Proposition 3.3 (Part 1), the route we take here is to show that every term on the LHS of eq. (17) is less than or equal to some unique terms appearing on the RHS. Again, we have terms on the LHS of two types (i) pairwise distances for matched elements (ii) penalisation of unmatched elements.

The argument for showing terms of the form (i) are less than or equal to some unique terms on the RHS is exactly the same as that in the proof of Proposition 3.3 (Part 1), relying on the fact that $d(\cdot, \cdot)$ is a metric and so obeys the triangle inequality. For brevity, we do not repeat this here.

The argument for terms of the form (ii) is slightly different and so we give full exposition. Considering first the penalisation terms for elements of X not included in the matching \mathcal{M}_3 , we will have a ρ for each $x \in (\mathcal{M}_3)_X^c$. We now seek to show that each ρ is less than or equal to some unique terms appearing on the RHS of eq. (17). For x to not be in \mathcal{M}_3 one of two things must have happened

1. $(x, z) \in \mathcal{M}_1^*$ for some $z \in Z$ with $(z, y) \notin \mathcal{M}_2^*$ for any $y \in Y$

$$\implies \text{a term on the RHS of } d(x, z) + \rho$$

which will also be unique to the pair (x, z) . Now, since $d(\cdot, \cdot)$ is a metric it is non-negative, thus

$$\rho \leq d(x, z) + \rho$$

as desired;

2. Alternatively, we might have $(x, z) \notin \mathcal{M}_1^*$ for any $z \in Z$

$$\implies \text{a term on the RHS of } \rho,$$

and thus in this case we trivially have

$$\rho \leq \rho.$$

In either case, we again have a term on the LHS of eq. (17) which is less than or equal to some unique terms on the RHS. Moreover, thus argument will apply similarly to the penalisation terms for Y without loss of generality. Thus we have that every term on the LHS of eq. (17) is less than or equal to some unique terms on the RHS, proving the inequality of eq. (17) holds. Thus $d_{M,\rho}$ satisfies condition (iii), completing the proof. \square

PROOF OF PROPOSITION 3.4. Given two multisets X and Y , let

$$C(\mathcal{M}) = \sum_{(x,y) \in \mathcal{M}} d(x, y) + \sum_{x \notin \mathcal{M}_X} d(x, \Lambda) + \sum_{y \notin \mathcal{M}_Y} d(y, \Lambda)$$

and, towards proving this result, we assume that any matching \mathcal{M}' for which

$$C(\mathcal{M}') = \min_{\mathcal{M}} C(\mathcal{M}) = d_M(X, Y),$$

is *not* complete, seeking a contradiction. There may be more than one such matching, so without loss of generality, let \mathcal{M}' denote any one of these optimal matchings. Since \mathcal{M}' is not complete, there must be a currently un-matched pair, that is, (x^*, y^*) such that $x^* \in X$ and $y^* \in Y$ but $x^* \notin \mathcal{M}'_X$ and $y^* \notin \mathcal{M}'_Y$. One can now define a new matching \mathcal{M}'' by augmenting \mathcal{M}' as follows

$$\mathcal{M}'' = \mathcal{M}' \cup \{(x^*, y^*)\}$$

for which

$$\begin{aligned}
C(\mathcal{M}'') &= \sum_{(x,y) \in \mathcal{M}''} d(x,y) + \sum_{x \notin \mathcal{M}''_X} d(x, \Lambda) + \sum_{y \notin \mathcal{M}''_Y} d(y, \Lambda) \\
&= \sum_{(x,y) \in \mathcal{M}'} d(x,y) + d(x^*, y^*) + \sum_{x \notin \mathcal{M}''_X} d(x, \Lambda) + \sum_{y \notin \mathcal{M}''_Y} d(y, \Lambda) \\
(18) \quad &\leq \sum_{(x,y) \in \mathcal{M}'} d(x,y) + d(x^*, \Lambda) + d(y^*, \Lambda) + \sum_{x \notin \mathcal{M}''_X} d(x, \Lambda) + \sum_{y \notin \mathcal{M}''_Y} d(y, \Lambda) \\
&= \sum_{(x,y) \in \mathcal{M}'} d(x,y) + \sum_{x \notin \mathcal{M}'_X} d(x, \Lambda) + \sum_{y \notin \mathcal{M}'_Y} d(y, \Lambda) \\
&= C(\mathcal{M}')
\end{aligned}$$

where in the third line we use the fact $d(\cdot, \cdot)$ is a distance metric, and hence obeys the triangle inequality. Since \mathcal{M}' was optimal, we must also have $C(\mathcal{M}') \leq C(\mathcal{M})$ for all matchings \mathcal{M} , which combined with eq. (18) implies $C(\mathcal{M}'') = C(\mathcal{M}')$, that is, \mathcal{M}'' is also an optimal matching. Moreover, we have $|\mathcal{M}''| = |\mathcal{M}'| + 1$. Now, either (i) \mathcal{M}'' is complete, or (ii) we can repeat this augmentation, increasing the matching cardinality until it is complete. Either way, we arrive at a matching which is both optimal and complete, contradicting our assumption that all optimal matchings were not complete. The result now follows by contradiction. \square

PROOF OF PROPOSITION 3.5. Given two multisets X and Y , let

$$C(\mathcal{M}) = \sum_{(x,y) \in \mathcal{M}} d(x,y) + \rho(n + m - 2|\mathcal{M}|)$$

where $|X| = n$ and $|Y| = m$, and letting

$$K = \max_{x \in X, y \in Y} d(x, y)$$

we assume $\rho \geq K/2$. Towards proving the result, further assume any matching \mathcal{M}' for which

$$C(\mathcal{M}') = \min_{\mathcal{M}} C(\mathcal{M}) = d_{M,\rho}(X, Y)$$

is *not* complete, seeking a contradiction. As in the proof of Proposition 3.4, without loss of generality we let \mathcal{M}' denote one of these optimal matchings and define a new matching \mathcal{M}'' by augmenting \mathcal{M}' with a presently un-matched pair (x^*, y^*) , that is

$$\mathcal{M}'' = \mathcal{M}' \cup \{(x^*, y^*)\}$$

for which

$$\begin{aligned}
C(\mathcal{M}'') &= \sum_{(x,y) \in \mathcal{M}''} d(x,y) + \rho(n+m-2|\mathcal{M}''|) \\
&= \sum_{(x,y) \in \mathcal{M}'} d(x,y) + d(x^*,y^*) - 2\rho + \rho(n+m-2|\mathcal{M}'|) \\
(19) \quad &\leq \sum_{(x,y) \in \mathcal{M}'} d(x,y) + 2\rho - 2\rho + \rho(n+m-2|\mathcal{M}'|) \\
&= \sum_{(x,y) \in \mathcal{M}'} d(x,y) + \rho(n+m-2|\mathcal{M}'|) \\
&= C(\mathcal{M}')
\end{aligned}$$

where in the second line we use the fact that $|\mathcal{M}''| = |\mathcal{M}'| + 1$, whilst in the third line we used the fact that

$$d(x,y) \leq K \leq 2\rho$$

for any $x \in X$ and $y \in Y$, by definition of K and the assumption regarding ρ . As in the proof of Proposition 3.4, since \mathcal{M}' was assumed optimal, eq. (19) implies that \mathcal{M}'' must also be optimal. Moreover, either (i) \mathcal{M}'' is complete, or (ii) we may repeat this augmentation until it is. In either case, we arrive at a matching which is optimal and complete. Hence the result follows by contradiction. \square

PROOF OF PROPOSITION 3.7. In what follows we will use the notation $d_{W_1}(\mu_X, \mu_Y)$ for the 1-Wasserstein distances between the *distributions* μ_X and μ_Y , which is known to be a distance metric (Peyré and Cuturi, 2019, Prop. 2.2). Observe that by our definition of the EMD between multisets (Definition 3.6) we have $d_{\text{EMD}}(X, Y) = d_{W_1}(\mu_X, \mu_Y)$.

The conditions (ii) and (iii) are inherited naturally. Firstly, we have

$$\begin{aligned}
d_{\text{EMD}}(X, Y) &= d_{W_1}(\mu_X, \mu_Y) \\
&= d_{W_1}(\mu_Y, \mu_X) \\
&= d_{\text{EMD}}(Y, X)
\end{aligned}$$

where the second line follows since d_{W_1} is a metric between distributions, verifying that (ii) holds. Secondly, for any multisets X, Y and Z we have

$$\begin{aligned}
d_{\text{EMD}}(X, Y) &= d_{W_1}(\mu_X, \mu_Y) \\
&\leq d_{W_1}(\mu_X, \mu_Z) + d_{W_1}(\mu_Z, \mu_Y) \\
&= d_{\text{EMD}}(X, Z) + d_{\text{EMD}}(Z, Y)
\end{aligned}$$

where again the second line follows since d_{W_1} is a metric. Thus (iii) also holds.

We now assume metric condition (i) holds, seeking a contradiction. To do so, let X be a multiset and define Y via its multiplicity function as follows (for any $x \in \mathcal{X}$)

$$m_Y(x) = C \cdot m_X(x)$$

where $C \in \mathbb{Z}_+$, that is, Y and X are proportional. Observe that if $C > 1$ then $X \neq Y$ whilst

$$\mu_Y(x) = \frac{m_Y(x)}{|Y|} = \frac{C \cdot m_X(x)}{C \cdot |X|} = \mu_X(x)$$

for any $x \in \mathcal{X}$, that is, $\mu_X = \mu_Y$. Consequently, we have $X \neq Y$ and

$$d_{\text{EMD}}(X, Y) = d_{W_1}(\mu_X, \mu_Y) = 0,$$

thus contradicting the assumption condition (i) holds. \square

PROOF OF PROPOSITION 3.9. Firstly, since both d_{EMD} and d_s satisfy metric conditions (ii) and (iii), so will a linear combination thereof.

As in the proof of Proposition 3.7, we use the notation $d_{W_1}(\mu_X, \mu_Y)$ for the 1-Wasserstein distance between the *distributions* μ_X and μ_Y , known to be a distance metric (Peyré and Cuturi, 2019, Prop. 2.2). Furthermore, by our definition of the EMD between multisets (Definition 3.6) we have $d_{\text{EMD}}(X, Y) = d_{W_1}(\mu_X, \mu_Y)$.

Towards proving (i) holds, assume that $X = Y$, which implies $\mu_X = \mu_Y$ and $|X| = |Y|$. Since both d_{W_1} and d_s are metrics this implies $d_{W_1}(\mu_X, \mu_Y) = d_s(|X|, |Y|) = 0$, and so

$$d_{s\text{EMD}}(X, Y) = \tau \cdot 0 + (1 - \tau) \cdot 0 = 0.$$

Conversely, assume that $d_{s\text{EMD}}(X, Y) = 0$. Being a linear combination of non-negative terms, this implies both $d_{\text{EMD}}(X, Y) = 0$, and $d_s(|X|, |Y|) = 0$. Now, since d_s is a metric we have $|X| = |Y|$, whilst since $d_{\text{EMD}}(X, Y) = d_{W_1}(\mu_X, \mu_Y)$ we have $d_{W_1}(\mu_X, \mu_Y) = 0$ which, since d_{W_1} is a metric, implies $\mu_X = \mu_Y$, which together imply for any $x \in \mathcal{X}$ we must have

$$\begin{aligned} \mu_X(x) &= \mu_Y(x) \\ \implies \frac{m_X(x)}{|X|} &= \frac{m_Y(x)}{|Y|} \\ \implies m_X(x) &= m_Y(x) \quad (\text{since } |X| = |Y|) \end{aligned}$$

and hence $m_X = m_Y$, that is, $X = Y$. This confirms (i) and completes the proof. \square

B.2. Sequence distances.

REMARK. Both d_E and $d_{E,\rho}$ can be seen as special cases of the so-called *string edit distance* proposed by Wagner and Fischer (1974). We could, therefore, conclude right away that both are indeed distance metrics. However, for completeness, and to emphasise the close connections with the matching distances, we proceed to prove these results, emulating the structure and approach in the proof of Proposition 3.3.

PROOF OF PROPOSITION 4.3 (PART 1). To aide this exposition, we write $d_E(X, Y)$ in terms of its cost function as follows

$$d_E(X, Y) = \min_{\mathcal{M}} C(\mathcal{M})$$

where \mathcal{M} denotes a monotone matching of X and Y and

$$C(\mathcal{M}) = \sum_{(x,y) \in \mathcal{M}} d(x, y) + \sum_{x \in \mathcal{M}_X^c} d(x, \Lambda) + \sum_{y \in \mathcal{M}_Y^c} d(y, \Lambda)$$

denotes the cost of the matching \mathcal{M} .

We first consider metric condition (i) (identity of indiscernibles). Firstly, since d is a metric we have $d(x, y) \geq 0$, which implies $d_E(X, Y) \geq 0$ for any sequences X and Y . Now, assuming that $X = Y$, then letting $n = |X| = |Y|$ this implies

$$x_i = y_i \quad \text{for } i = 1, \dots, n.$$

Consequently, we can trivially construct a monotone matching \mathcal{M}^* which pairs equivalent entries

$$(20) \quad \mathcal{M}^* = \{(x_1, y_1), \dots, (x_n, y_n)\}.$$

The existence of this matching thus leads to an upper bound on the distance

$$\begin{aligned} d_E(X, Y) &\leq C(\mathcal{M}^*) \\ &= \sum_{i=1}^n d(x_i, y_i) + 0 + 0 = 0, \end{aligned}$$

which, combined with $d_E(X, Y) \geq 0$, implies $d_E(X, Y) = 0$. Conversely, assume that $d_E(X, Y) = 0$. This implies that both penalisation terms are zero, and therefore every entry of each sequence is included in the matching. Furthermore, this implies the sequences are of equal length, that is, $|X| = |Y|$. The only possible *monotone* matching which includes all sequence entries is the \mathcal{M}^* seen in eq. (20), thus

$$d_E(X, Y) = \sum_{i=1}^n d(x_i, y_i) = 0$$

which, since $d(x, y) \geq 0$, implies

$$d(x_i, y_i) = 0 \quad (\text{for } i = 1, \dots, n).$$

since d is a metric it satisfies condition (i), and thus we have

$$x_i = y_i \quad (\text{for } i = 1, \dots, n)$$

that is, $X = Y$, confirming that d_E satisfies (i).

The symmetry condition (ii) follows trivially from the symmetry of $d(x, y)$ and the penalisation terms.

We now finish with confirming the triangle inequality (iii) is satisfied. The approach is almost identical to the proof of Proposition 3.3 (Part 1), with one key difference: we must ensure all matchings are monotone. Assuming that $X = (x_1, \dots, x_n)$, $Y = (y_1, \dots, y_m)$ and $Z = (z_1, \dots, z_l)$ are three sequences, we seek to show that

$$d_E(X, Y) \leq d_E(X, Z) + d_E(Z, Y).$$

With \mathcal{M}_1^* and \mathcal{M}_2^* denoting optimal monotone matchings for $d_E(X, Z)$ and $d_E(Z, Y)$ respectively, that is

$$d_E(X, Z) = C(\mathcal{M}_1^*) \quad d_E(Z, Y) = C(\mathcal{M}_2^*)$$

these induce the following matching \mathcal{M}_3 of X and Y

$$(21) \quad \mathcal{M}_3 = \{(x_i, y_j) : (x_i, z_k) \in \mathcal{M}_1^* \text{ and } (z_k, y_j) \in \mathcal{M}_2^* \text{ for some } z_k \in Z\}$$

that is, we match entries of X and Y if they were matched to the same entry of Z .

We now confirm this is a monotone matching. Recall that \mathcal{M}_3 is monotone if for any pairs (x_{i_1}, y_{j_1}) and (x_{i_2}, y_{j_2}) in \mathcal{M}_3 we have

$$i_1 < i_2 \iff j_1 < j_2.$$

By the definition of \mathcal{M}_3 there exists z_{k_1} and z_{k_2} in Z such that

$$\begin{aligned} (x_{i_1}, z_{k_1}) \in \mathcal{M}_1^* & \quad (z_{k_1}, y_{j_1}) \in \mathcal{M}_2^* \\ (x_{i_2}, z_{k_2}) \in \mathcal{M}_1^* & \quad (z_{k_2}, y_{j_2}) \in \mathcal{M}_2^* \end{aligned}$$

Furthermore, since \mathcal{M}_1^* and \mathcal{M}_2^* are monotone we have

$$i_1 < i_2 \iff k_1 < k_2 \quad \text{and} \quad k_1 < k_2 \iff j_1 < j_2$$

which therefore implies

$$i_1 < i_2 \iff k_1 < k_2 \iff j_1 < j_2$$

and hence \mathcal{M}_3 is also monotone. Now that we have shown the induced matching is indeed monotone, observe that by definition we have the following

$$d_E(X, Y) \leq C(\mathcal{M}_3)$$

which implies the triangle inequality will hold if we can show the following inequality is satisfied

$$(22) \quad C(\mathcal{M}_3) \leq d_E(X, Z) + d_E(Z, Y).$$

The argument for this is identical to that used to show eq. (16) in the proof of Proposition 3.3 (Part 1). For brevity, we do not repeat the steps and henceforth assume eq. (22) holds. Thus the triangle inequality (iii) holds, completing the proof. \square

PROOF OF PROPOSITION 4.3 (PART 2). To aide this exposition, we write $d_{E,\rho}(X, Y)$ in terms of its cost function as follows

$$d_{E,\rho}(X, Y) = \min_{\mathcal{M}} C(\mathcal{M})$$

where \mathcal{M} denotes a monotone matching of X and Y and

$$C(\mathcal{M}) = \sum_{(x,y) \in \mathcal{M}} d(x, y) + \rho(|X| + |Y| - 2|\mathcal{M}|)$$

denotes the cost of the matching \mathcal{M} .

Conditions (i) and (ii) can be shown to hold by following the same reasoning seen in the proof of Proposition 4.3 (Part 1). For brevity, we therefore do not repeat the details. We will, however, outline the argument confirming satisfaction of the triangle inequality (iii).

Assuming $X = (x_1, \dots, x_n)$, $Y = (y_1, \dots, y_m)$ and $Z = (z_1, \dots, z_l)$ are sequences, we are looking to show

$$d_{E,\rho}(X, Y) \leq d_{E,\rho}(X, Z) + d_{E,\rho}(Z, Y).$$

Letting \mathcal{M}_1^* and \mathcal{M}_2^* denote the optimal monotone matchings of $d_{E,\rho}(X, Z)$ and $d_{E,\rho}(Z, Y)$ respectively, that is

$$d_{E,\rho}(X, Z) = C(\mathcal{M}_1^*) \quad d_{E,\rho}(Z, Y) = C(\mathcal{M}_2^*)$$

we can find a monotone matching \mathcal{M}_3 of X and Y induced by \mathcal{M}_1^* and \mathcal{M}_2^* as was done in the proof of Proposition 4.3 (Part 1), that is, take \mathcal{M}_3 as in eq. (21), where we matched entries of X and Y if they were matched to the same entry of Z .

The next step in proving (iii) is to confirm the following inequality holds

$$(23) \quad C(\mathcal{M}_3) \leq d_{E,\rho}(X, Z) + d_{E,\rho}(Z, Y),$$

for which we again appeal arguments in a previous proof. Specifically, that of Proposition 3.3 (Part 2), where a similar inequality was shown to hold for the fixed penalty matching distance, namely that of eq. (17). Recall the argument therein used properties of \mathcal{M}_3 to show that every term on the LHS is less than or equal to some unique terms on the RHS. Since the same \mathcal{M}_3 has been taken in the present case, the argument can also be applied here. As such, it will henceforth be assumed eq. (23) holds. Thus the triangle inequality (iii) is satisfied, completing the proof. \square

PROOF OF PROPOSITION 4.5. For ease of reference, recall the DTW distance between sequences X and Y is given by the following

$$d_{\text{DTW}}(X, Y) = \min_{\mathcal{C}} \left\{ \sum_{(x, y) \in \mathcal{C}} d(x, y) \right\}$$

where \mathcal{C} is a coupling.

Observe the symmetry condition (ii) follows trivially from the symmetry of the ground distance $d(\cdot, \cdot)$, by virtue of it being a metric.

We now show that conditions (i) and (iii) are violated by providing counterexamples. Beginning with (i), consider the following two sequences

$$\begin{aligned} X &= (x_1) & Y &= (y_1, y_2) \\ &= (\tilde{x}) & &= (\tilde{x}, \tilde{x}) \end{aligned}$$

where $\tilde{x} \in \mathcal{X}$ denotes an arbitrary element of the underlying space. Clearly, we have $X \neq Y$. However, there is only one valid coupling of X and Y , namely $\mathcal{C} = ((x_1, y_1), (x_1, y_2))$. Consequently, the DTW distance is given by

$$\begin{aligned} d_{\text{DTW}}(X, Y) &= \sum_{(x, y) \in \mathcal{C}} d(x, y) \\ &= d(x_1, y_1) + d_I(x_1, y_2) \\ &= d(\tilde{x}, \tilde{x}) + d(\tilde{x}, \tilde{x}) = 0 + 0 \end{aligned}$$

violating condition (i).

Turning now to condition (iii), consider the following three sequences

$$(24) \quad \begin{aligned} X &= (x_1, x_2) & Z &= (z_1) & Y &= (y_1) \\ &= (\tilde{x}, \tilde{x}) & &= (\tilde{x}) & &= (\tilde{y}) \end{aligned}$$

where $\tilde{x}, \tilde{y} \in \mathcal{X}$ with $\tilde{x} \neq \tilde{y}$. Now, the only valid coupling of X and Z is given by $\mathcal{C}_{XZ} = ((x_1, z_1), (x_2, z_1))$, similarly the only coupling of Z and Y is given by $\mathcal{C}_{ZY} = ((z_1, y_1))$, whilst for X and Y this will be $\mathcal{C}_{XY} = ((x_1, y_1), (x_2, y_1))$. This therefore implies

$$\begin{aligned} d_{\text{DTW}}(X, Y) &= \sum_{(x, y) \in \mathcal{C}_{XY}} d(x, y) \\ &= d(x_1, y_1) + d(x_2, y_1) \\ &= d(\tilde{x}, \tilde{y}) + d(\tilde{x}, \tilde{y}) \\ &= 2d(\tilde{x}, \tilde{y}) \end{aligned}$$

whilst

$$\begin{aligned} d_{\text{DTW}}(X, Z) + d_{\text{DTW}}(Z, Y) &= \sum_{(x, z) \in \mathcal{C}_{XZ}} d(x, z) + \sum_{(z, y) \in \mathcal{C}_{ZY}} d(z, y) \\ &= [d(x_1, z_1) + d(x_2, z_1)] + [d(z_1, y_1)] \\ &= [d(\tilde{x}, \tilde{x}) + d(\tilde{x}, \tilde{x})] + [d(\tilde{x}, \tilde{y})] \\ &= d(\tilde{x}, \tilde{y}) \end{aligned}$$

where we have used the fact, since d is a metric, we have $d(x, x) = 0$. Now, since $\tilde{x} \neq \tilde{y}$ we have $d(\tilde{x}, \tilde{y}) > 0$, implying

$$d_{\text{DTW}}(X, Y) = 2d(\tilde{x}, \tilde{y}) > d(\tilde{x}, \tilde{y}) = d_{\text{DTW}}(X, Z) + d_{\text{DTW}}(Z, Y)$$

and (iii) is violated, as desired. This completes the proof. \square

PROOF OF PROPOSITION 4.7. For ease of reference, recall the fixed-penalty DTW distance between sequences X and Y is given by the following

$$(25) \quad d_{\text{DTW},\rho}(X, Y) = \min_{\mathcal{C}} \left\{ \sum_{(x,y) \in \mathcal{C}} d(x, y) + \rho \cdot w(\mathcal{C}) \right\}$$

where

$$w(\mathcal{C}) := |\{(x_i, y_j) \in \mathcal{C} : (x_i, y_{j+1}) \in \mathcal{C} \text{ or } (x_{i+1}, y_j) \in \mathcal{C}\}|,$$

quantifies the amount of warping in \mathcal{C} , whilst $\rho > 0$ is a parameter controlling the penalisation incurred for each instance of warping.

We first show that condition (i) (identity of indiscernibles) holds. To do so, let X and Y be two sequences such that $d_{\text{DTW},\rho}(X, Y) = 0$. Assuming that \mathcal{C}^* denotes an optimal coupling, that is

$$d_{\text{DTW},\rho}(X, Y) = \sum_{(x,y) \in \mathcal{C}^*} d(x, y) + \rho \cdot w(\mathcal{C}^*)$$

implying

$$(26) \quad \sum_{(x,y) \in \mathcal{C}^*} d(x, y) + \rho \cdot w(\mathcal{C}^*) = 0,$$

where we here use the fact that $d_{\text{DTW},\rho}(X, Y) = 0$. Notice since $d(x, y) \geq 0$, $w(\mathcal{C}) \geq 0$ and $\rho > 0$ this implies each term in eq. (26) must be zero. In particular, we must have $w(\mathcal{C}^*) = 0$, that is, no warping has taken place. Thus, each entry of X is paired with exactly one from Y . Observe this implies $|X| = |Y| = N$ and furthermore the only coupling possible is the following

$$\mathcal{C}^* = \{(x_i, y_i) : i = 1, \dots, N\}$$

that is, we pair the first entries, second entries, and so on. Moreover, due to eq. (26) the sum of pairwise distances in \mathcal{C}^* must be zero, implying

$$\begin{aligned} d_{\text{DTW},\rho}(X, Y) &= \sum_{(x,y) \in \mathcal{C}^*} d(x, y) \\ &= \sum_{i=1}^N d(x_i, y_i) = 0 \end{aligned}$$

and now since d is a metric we have $d(x_i, y_i) \geq 0$ this implies $d(x_i, y_i) = 0$ for $i = 1, \dots, N$. Finally, using again the fact d is a metric this implies $x_i = y_i$ for $i = 1, \dots, N$ and consequently we have $X = Y$.

Conversely, assume that X and Y are sequences such that $X = Y$. With \mathcal{C}^* again denoting the coupling obtained by pairing the i th entry of X with the i th entry of Y , that is

$$\mathcal{C}^* = \{(x_i, y_i) : i = 1, \dots, N\}$$

where $N = |X| = |Y|$, this implies the following upper bound

$$\begin{aligned} d_{\text{DTW},\rho}(X, Y) &\leq \sum_{(x,y) \in \mathcal{C}^*} d(x, y) + \rho \cdot w(\mathcal{C}^*) \\ &= \sum_{i=1}^N d(x_i, y_i) \\ &= 0 \end{aligned}$$

where in the second line we use the fact $w(\mathcal{C}^*) = 0$, whilst the third line follows since $x_i = y_i$ for $i = 1, \dots, N$ by assumption and d is a metric. Finally, since $d_{\text{DTW},\rho}(X, Y) \geq 0$ by definition, this implies we must have $d_{\text{DTW},\rho}(X, Y) = 0$. This confirms that condition (i) holds.

The symmetry condition (ii) follows trivially from the symmetry of the ground distance d and the penalisation term.

We now show the triangle inequality (iii) is violated. Here as a counterexample we consider the sequence X, Y and Z defined in eq. (24) as seen in the proof of Proposition 4.5, with $\mathcal{C}_{XZ}, \mathcal{C}_{ZY}$ and \mathcal{C}_{XY} the associated couplings. As in the proof of Proposition 4.5, this implies

$$\begin{aligned} d_{\text{DTW},\rho}(X, Y) &= \sum_{(x,y) \in \mathcal{C}_{XY}} d(x, y) + \rho \cdot w(\mathcal{C}_{XY}) \\ &= d(x_1, y_1) + d(x_2, y_1) + \rho \\ &= d(\tilde{x}, \tilde{y}) + d(\tilde{x}, \tilde{y}) + \rho \\ &= 2d(\tilde{x}, \tilde{y}) + \rho \end{aligned}$$

whilst

$$\begin{aligned} d_{\text{DTW},\rho}(X, Z) + d_{\text{DTW},\rho}(Z, Y) &= \left(\sum_{(x,z) \in \mathcal{C}_{XZ}} d(x, z) + \rho \cdot w(\mathcal{C}_{XZ}) \right) \\ &\quad + \left(\sum_{(z,y) \in \mathcal{C}_{ZY}} d(z, y) + \rho \cdot w(\mathcal{C}_{ZY}) \right) \\ &= [d(x_1, z_1) + d(x_2, z_1)] + [d(z_1, y_1)] \\ &= [d(\tilde{x}, \tilde{x}) + d(\tilde{x}, \tilde{x}) + \rho] + [d(\tilde{x}, \tilde{y})] \\ &= d(\tilde{x}, \tilde{y}) + \rho \end{aligned}$$

where we have used the fact, since d is a metric, we have $d(x, x) = 0$. Now, since $\tilde{x} \neq \tilde{y}$ we have $d(\tilde{x}, \tilde{y}) > 0$, implying

$$d_{\text{DTW},\rho}(X, Y) = 2d(\tilde{x}, \tilde{y}) + \rho > d(\tilde{x}, \tilde{y}) + \rho = d_{\text{DTW},\rho}(X, Z) + d_{\text{DTW},\rho}(Z, Y)$$

thus violating (iii), as desired. This completes the proof. \square

APPENDIX C: PATH DISTANCES

In this section, we provide further details regarding two path distances, one of which was invoked in the data analysis of Section 5. Both are defined via maximally-sized substructures shared by the two paths being compared, considering in particular common subsequences and subpaths (Figure 4).

For a path $x = (x_1, \dots, x_n)$, where each $x_i \in \mathcal{V}$ with \mathcal{V} some vertex set, we denote a *subpath* of x from index i to j by the following

$$x_{i:j} = (x_i, \dots, x_j)$$

where $1 \leq i \leq j \leq n$ (Figure 4a). More generally, assume that $\mathbf{v} = (v_1, \dots, v_s)$ with $1 \leq v_1 < v_2 < \dots < v_s \leq n$, then a *subsequence* of x can be obtained by indexing with \mathbf{v} as follows

$$x_{\mathbf{v}} = (x_{v_1}, \dots, x_{v_s})$$

which will be of length s (Figure 4b). Observe that every subpath of x is also a subsequence, making a subsequence the more general of the two structures.

Given another path $y = (y_1, \dots, y_m)$, one also can consider the notion of *common* subpaths and subsequences. Namely, a common subpath of x and y occurs when we have

$$x_{i:j} = y_{l:k}$$

for some $1 \leq i \leq j \leq n$ and $1 \leq l \leq k \leq m$. Similarly, a common subsequence of x and y occurs when

$$x_v = y_u$$

for some $1 \leq v_1 < v_2 < \dots < v_s \leq n$ and $1 \leq u_1 < u_2 < \dots < u_s \leq m$.

The more similar x and y are the larger we might expect their common subpaths or subsequences to be. Following this rationale, one can define distances between x and y by finding common subpaths or subsequences which *maximal*, that is, one for which there exist no other common subpaths or subsequences of greater length. This leads to the following definitions.

DEFINITION C.1 (Longest common subsequence distance). For two paths $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_m)$ the longest common subsequence (LCS) distance is given by the following

$$d_{\text{LCS}}(x, y) := n + m - 2\delta_{\text{LCS}}$$

where

$$\delta_{\text{LCS}} = \max\{|\mathbf{v}| = |\mathbf{u}| : x_{\mathbf{v}} = y_{\mathbf{u}}\},$$

where $|\mathbf{v}|$ denotes the length of \mathbf{v} , so that δ_{LSP} denotes the maximum length subsequence common to both x and y .

DEFINITION C.2 (Longest common subpath distance). For two paths $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_m)$ the longest common subpath (LSP) distance is given by the following

$$d_{\text{LSP}}(x, y) := n + m - \delta_{\text{LSP}}$$

where

$$\delta_{\text{LSP}} = \max\{|i : j| = |l : k| : x_{i:j} = y_{l:k}\},$$

denoting the maximum length subpath common to both x and y .

Both the LCS and LSP distances can be shown to be metrics, that is, they satisfy all three metric conditions, as we summarise via the following result.

PROPOSITION C.3. *Both d_{LCS} and d_{LSP} satisfy metric conditions (i)-(iii).*

PROOF OF PROPOSITION C.3. Consider first condition (i) (identity of indiscernibles). If we have two paths $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_m)$ such that $d_{\text{LCS}}(x, y) = 0$, this implies

$$(27) \quad n + m - 2\delta_{\text{LCS}} = 0.$$

Moreover, by definition $\delta_{\text{LCS}} \leq \min(n, m)$, since a common subsequence cannot be longer than the shorter path. We claim that this implies $n = m$. Towards doing so, if we assume $n < m$ this implies

$$n + m > 2n \geq 2\delta_{\text{LCS}}$$

where we have used the fact $\delta_{\text{LCS}} \leq n$, which contradicts eq. (27). A similar contradiction can be made if we assume $n > m$, and consequently we must have $n = m$. Substituting this into eq. (27) leads to $\delta_{\text{LCS}} = n = m$ which implies that x and y share a common subsequence of the same length as themselves, that is $x = y$. Conversely, if $x = y$ then it should be clear that the maximum common subsequence will be the one including all their entries, that is $\delta_{\text{LCS}} = n = m$ and hence

$$d_{\text{LCS}}(x, y) = n + m - 2\delta_{\text{LCS}} = 0.$$

This proves that (i) holds for the LCS distance, and an identical argument can be used to show it similarly holds for the LSP distance.

The symmetry condition (ii) for both the LCS and LSP distances follows trivially from the symmetry in the definition of common subsequences and subpaths, respectively.

Finally we turn to the triangle inequality (iii), considering first the LCS distance. Assume that $x = (x_1, \dots, x_n)$, $y = (y_1, \dots, y_m)$ and $z = (z_1, \dots, z_k)$ are three paths and that δ_{xz} , δ_{zy} and δ_{xy} are such that

$$d_{\text{LCS}}(x, y) = n + m - 2\delta_{xy} \quad d_{\text{LCS}}(x, z) = n + k - 2\delta_{xz} \quad d_{\text{LCS}}(z, y) = n + m - 2\delta_{zy},$$

then, assuming the triangle inequality holds, we have

$$d_{\text{LCS}}(x, y) \leq d_{\text{LCS}}(x, z) + d_{\text{LCS}}(z, y)$$

which is equivalent to the following

$$n + m - 2\delta_{xy} \leq n + k - 2\delta_{xz} + n + m - 2\delta_{zy}$$

which is true if and only if

$$(28) \quad \delta_{xz} + \delta_{zy} - k \leq \delta_{xy}.$$

Notice that if eq. (28) holds then one can trace the implications back to conclude the triangle inequality also holds. Towards doing so, we consider a finding the common subsequence between x and y induced by those between x and z and y and z , which will allow us to obtain the desired lower bound.

To aide this exposition we introduce some notation. In particular, for two subsequences v and u of $[s] = (1, \dots, s)$ we can extend the notion of unions and intersections used for sets, that is $v \cup u$ and $v \cap u$ respectively, where if $w = v \cap u$ then each entry thereof w_i appears in both v and u , whilst if $w = v \cup u$ then each w_i appears in at least one of u and v . Moreover, with $|v|$ denoting the length of subsequence v , as for sets the following identity will hold

$$|v| + |u| - |v \cap u| = |v \cup u|.$$

Now suppose that v_{xz} and v_{zy} index a maximal common subsequences of z with x and y respectively, that is, both are subsequences of $[k]$ such that

$$xv_{xz} = zv_{xz} \quad zv_{zy} = yv_{zy}$$

for some subsequences v_{xz} of $[n]$ and v_{zy} of $[m]$, where $|v_{xz}| = \delta_{xz}$ and $|v_{zy}| = \delta_{zy}$. Notice now the entries of z indexed by $v_{xz} \cap v_{zy}$ will induce a common subsequence of x and y (by considering the associated entries of each). As such, if we let $\delta^* = |v_{xz} \cap v_{zy}|$ we will have

$$\delta^* \leq \delta_{xy}$$

by virtue of δ_{xy} being the *maximal* length of a common subsequence between x and y . Also by the inclusion-exclusion-like identity above we will have

$$\delta_{xz} + \delta_{zy} - \delta^* = |v_{xz} \cup v_{zy}| \leq k$$

where the inequality here follows since u_{zx} and v_{zy} are subsequences of $[k]$ (since they index z , which is of length k). Combining these last two inequalities thus leads to the following

$$\delta_{xz} + \delta_{zy} - k \leq \delta^* \leq \delta_{xy},$$

hence confirming eq. (28) holds and proving the triangle inequality holds for the LCS distance.

A similar argument can be used to prove that condition (iii) also holds for the LSP distance. For brevity we will not give full exposition here, but we note the only key difference will be the notion of intersections and unions. If we introduce the shorthand notation $(i : j) = (i, \dots, j)$ where $1 \leq i \leq j \leq s$, denoting the subpath of $[s]$ from i to j (notice this is consistent with notation used in Definition C.2), then we naturally have the following

$$(i : j) \cap (l : k) = (\max(i, l) : \min(j, k)) \quad (i : j) \cup (l : k) = (\min(i, l) : \max(j, k)),$$

and moreover if $|(i : j)| = j - i + 1$ denotes subpath length we will similarly have the following identity

$$|(i : j)| + |(l : k)| - |(i : j) \cap (l : k)| = |(i : j) \cup (l : k)|.$$

With these results one can follow the rationale used for the LCS distance, replacing subsequences with subpaths, to show that the triangle inequality is satisfied.

Thus conditions (i), (ii) and (iii) all hold for both the LCS and LSP distances, completing the proof. \square

Finally, we discuss computation. Both of these distances can be computed via dynamic programming, much like the edit and DTW distances (Appendices A.2 and A.3), with a time complexity of $\mathcal{O}(nm)$. Infact, the LCS distance can be seen as an instance of the fixed-penalty edit distance $d_{E,\rho}$ with ground distance given by

$$d(x_i, y_j) = \begin{cases} 0 & \text{if } x_i = y_j \\ 2 & \text{otherwise} \end{cases}$$

and $\rho = 1$. Consequently, one can apply Algorithm 3 or Algorithm 4 directly, substituting in these values for $d(\cdot, \cdot)$ and ρ .

The approach to evaluate d_{LSP} is slightly different. In this case, we essentially scan over x and y and keep track of the common subpaths seen. Formally, we construct an $n \times m$ matrix Q incrementally via the following recursive formula

$$Q_{(i+1)(j+1)} = \begin{cases} Q_{ij} + 1 & \text{if } x_i = y_j \\ 0 & \text{otherwise} \end{cases},$$

where when common subpaths appear between x and y one will see increments in Q diagonally. The maximum length of a subpath can thus be obtained by taking the element-wise maximum of Q , that is $\delta_{\text{LSP}} = \max_{ij} Q_{ij}$, which can then be plugged into Definition C.2 to compute $d_{\text{LSP}}(x, y)$. We summarise this in Algorithm 9, where we keep track of the maximum in Q as it is filled. Moreover, a lighter-memory algorithm is outlined in Algorithm 10, making use of the fact we only need to know the current and previous rows of Q .

APPENDIX D: PSEUDOCODE

Algorithm 1: Evaluating edit distance d_E **Data:** Sequences $X = (x_1, \dots, x_n)$ and $Y = (y_1, \dots, y_m)$ **Input:** $d(\cdot, \cdot)$ a distance metric between sequence entries**Result:** $d_E(X, Y)$ (Definition 4.1) $C \in \mathbb{R}^{(n+1) \times (m+1)}$; $C_{11} = 0$; $C_{(i+1)1} = C_{i1} + d(x_i, \Lambda)$ (for $i = 1, \dots, n$); $C_{1(j+1)} = C_{1j} + d(y_j, \Lambda)$ (for $j = 1, \dots, m$);**for** $i = 1, \dots, n$ **do** **for** $j = 1, \dots, m$ **do**

$$C_{(i+1)(j+1)} = \min \begin{cases} C_{ij} + d(x_i, y_j) \\ C_{(i+1)j} + d(y_j, \Lambda) \\ C_{i(j+1)} + d(x_i, \Lambda) \end{cases}$$

end**end****return** $C_{(n+1)(m+1)}$ **Algorithm 2:** Evaluating edit distance d_E (light memory)**Data:** Sequences $X = (x_1, \dots, x_n)$ and $Y = (y_1, \dots, y_m)$ **Input:** $d(\cdot, \cdot)$ a distance metric between sequence entries**Result:** $d_E(X, Y)$ (Definition 4.1) $Z^{\text{prev}}, Z^{\text{curr}} \in \mathbb{R}^{(m+1)}$; $Z_1^{\text{prev}} = 0, Z_1^{\text{curr}} = 0$; $Z_{i+1}^{\text{prev}} = Z_i^{\text{prev}} + d(y_i, \Lambda)$ (for $i = 1, \dots, m$);**for** $i = 1, \dots, n$ **do** $Z_1^{\text{curr}} = Z_1^{\text{curr}} + d(x_i, \Lambda)$; **for** $j = 1, \dots, m$ **do**

$$Z_{j+1}^{\text{curr}} = \min \begin{cases} Z_j^{\text{prev}} + d(x_i, y_j) \\ Z_{j+1}^{\text{prev}} + d(x_i, \Lambda) \\ Z_j^{\text{curr}} + d(y_j, \Lambda) \end{cases}$$

end $Z^{\text{prev}} = Z^{\text{curr}}$ **end****return** Z_{m+1}^{curr}

Algorithm 3: Evaluating fixed-penalty edit distance $d_{E,\rho}$ **Data:** Sequences $X = (x_1, \dots, x_n)$ and $Y = (y_1, \dots, y_m)$ **Input:** $d(\cdot, \cdot)$ a distance metric between sequence entries**Result:** $d_{E,\rho}(X, Y)$ (Definition 4.2) $C \in \mathbb{R}^{(n+1) \times (m+1)}$; $C_{(i+1)1} = i\rho$ (for $i = 0, \dots, n$); $C_{1(j+1)} = j\rho$ (for $j = 0, \dots, m$);**for** $i = 1, \dots, n$ **do** **for** $j = 1, \dots, m$ **do**

$$C_{(i+1)(j+1)} = \min \begin{cases} C_{ij} + d(x_i, y_j) \\ C_{i(j+1)} + \rho \\ C_{(i+1)j} + \rho \end{cases}$$

end**end****return** $C_{(n+1)(m+1)}$ **Algorithm 4:** Evaluating fixed-penalty edit distance $d_{E,\rho}$ (light memory)**Data:** Sequences $X = (x_1, \dots, x_n)$ and $Y = (y_1, \dots, y_m)$ **Input:** $d(\cdot, \cdot)$ a distance metric between sequence entries**Result:** $d_{E,\rho}(X, Y)$ (Definition 4.2) $Z^{\text{prev}}, Z^{\text{curr}} \in \mathbb{R}^{(m+1)}$; $Z_1^{\text{prev}} = 0, Z_1^{\text{curr}} = 0$; $Z_{i+1}^{\text{prev}} = Z_i^{\text{prev}} + \rho$ (for $i = 1, \dots, m$);**for** $i = 1, \dots, n$ **do** $Z_1^{\text{curr}} = Z_1^{\text{curr}} + d(x_i, \Lambda)$; **for** $j = 1, \dots, m$ **do**

$$Z_{j+1}^{\text{curr}} = \min \begin{cases} Z_j^{\text{prev}} + d(x_i, y_j) \\ Z_{j+1}^{\text{prev}} + \rho \\ Z_j^{\text{curr}} + \rho \end{cases}$$

end $Z^{\text{prev}} = Z^{\text{curr}}$ **end****return** Z_{m+1}^{curr}

Algorithm 5: Evaluating dynamic time warping distance d_{DTW}

Data: Sequences $X = (x_1, \dots, x_n)$ and $Y = (y_1, \dots, y_m)$
Input: $d(\cdot, \cdot)$ a distance metric between sequence entries
Result: $d_{\text{DTW}}(X, Y)$ (Definition 4.4)
 $C \in \mathbb{R}^{(n+1) \times (m+1)}$;
 $C_{11} = 0$;
 $C_{(i+1)1} = \infty$ (for $i = 1, \dots, n$);
 $C_{1(j+1)} = \infty$ (for $j = 1, \dots, m$);
for $i = 1, \dots, n$ **do**
 for $j = 1, \dots, m$ **do**
 $C_{(i+1)(j+1)} = d(x_i, y_j) + \min\{C_{ij}, C_{(i+1)j}, C_{i(j+1)}\}$
 end
end
return $C_{(n+1)(m+1)}$

Algorithm 6: Evaluating dynamic time warping distance d_{DTW} (light memory)

Data: Sequences $X = (x_1, \dots, x_n)$ and $Y = (y_1, \dots, y_m)$
Input: $d(\cdot, \cdot)$ a distance metric between sequence entries
Result: $d_{\text{DTW}}(X, Y)$ (Definition 4.4)
 $Z^{\text{prev}}, Z^{\text{curr}} \in \mathbb{R}^{(m+1)}$;
 $Z_1^{\text{prev}} = 0, Z^{\text{curr}} = \infty$;
 $Z_{i+1}^{\text{prev}} = \infty$ (for $i = 1, \dots, m$);
for $i = 1, \dots, n$ **do**
 for $j = 1, \dots, m$ **do**
 $Z_{j+1}^{\text{curr}} = d(x_i, y_j) + \min\{Z_j^{\text{prev}}, Z_j^{\text{curr}}, Z_{j+1}^{\text{prev}}\}$
 end
 $Z^{\text{prev}} = Z^{\text{curr}}$
end
return Z_{m+1}^{curr}

Algorithm 7: Evaluating fixed-penalty dynamic time warping distance $d_{\text{DTW},\rho}$

Data: Sequences $X = (x_1, \dots, x_n)$ and $Y = (y_1, \dots, y_m)$
Input: $d(\cdot, \cdot)$ a distance metric between sequence entries
Result: $d_{\text{DTW},\rho}(X, Y)$ (Definition 4.6)
 $C \in \mathbb{R}^{(n+1) \times (m+1)}$;
 $C_{11} = 0$;
 $C_{(i+1)1} = \infty$ (for $i = 1, \dots, n$);
 $C_{1(j+1)} = \infty$ (for $j = 1, \dots, m$);
for $i = 1, \dots, n$ **do**
 for $j = 1, \dots, m$ **do**
 $C_{(i+1)(j+1)} = d(x_i, y_j) + \min\{C_{ij}, C_{(i+1)j} + \rho, C_{i(j+1)} + \rho\}$
 end
end
return $C_{(n+1)(m+1)}$

Algorithm 8: Evaluating fixed-penalty dynamic time warping distance $d_{\text{DTW},\rho}$ (light memory)

Data: Sequences $X = (x_1, \dots, x_n)$ and $Y = (y_1, \dots, y_m)$
Input: $d(\cdot, \cdot)$ a distance metric between sequence entries
Result: $d_{\text{DTW},\rho}(X, Y)$ (Definition 4.6)
 $Z^{\text{prev}}, Z^{\text{curr}} \in \mathbb{R}^{(m+1)}$;
 $Z_1^{\text{prev}} = 0, Z_1^{\text{curr}} = \infty$;
 $Z_{i+1}^{\text{prev}} = \infty$ (for $i = 1, \dots, m$);
for $i = 1, \dots, n$ **do**
 for $j = 1, \dots, m$ **do**
 $Z_{j+1}^{\text{curr}} = d(x_i, y_j) + \min\{Z_j^{\text{prev}}, Z_j^{\text{curr}} + \rho, Z_{j+1}^{\text{prev}} + \rho\}$
 end
 $Z^{\text{prev}} = Z^{\text{curr}}$
end
return Z_{m+1}^{curr}

Algorithm 9: Evaluating LSP distance d_{LSP} **Data:** Paths $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_m)$ **Result:** $d_{\text{LSP}}(x, y)$ (Definition C.2) $Q \in \mathbb{Z}_+^{(n+1) \times (m+1)}$; $Q_{11} = 0$; $Q_{(i+1)1} = 0$ (for $i = 1, \dots, n$); $Q_{1(j+1)} = 0$ (for $j = 1, \dots, m$); $\delta = 0$;**for** $i = 1, \dots, n$ **do** **for** $j = 1, \dots, m$ **do** **if** $x_i = y_j$ **then** $Q_{(i+1)(j+1)} = Q_{ij} + 1$ $\delta = \max(z, Q_{(i+1)(j+1)})$ **else** $Q_{(i+1)(j+1)} = 0$ **end** **end****end****return** $n + m - 2\delta$ **Algorithm 10:** Evaluating LSP distance d_{LSP} (light memory)**Data:** Paths $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_m)$ **Result:** $d_{\text{LSP}}(x, y)$ (Definition C.2) $Z^{\text{prev}}, Z^{\text{curr}} \in \mathbb{Z}_+^{(m+1)}$; $Z_{i+1}^{\text{prev}} = Z_{i+1}^{\text{curr}} = 0$ (for $i = 0, \dots, m$); $\delta = 0$;**for** $i = 1, \dots, n$ **do** **for** $j = 1, \dots, m$ **do** **if** $x_i = y_j$ **then** $Z_{j+1}^{\text{curr}} = Z_j^{\text{prev}} + 1$ $\delta = \max(z, Z_{j+1}^{\text{curr}})$ **else** $Z_{j+1}^{\text{curr}} = 0$ **end** **end** $Z^{\text{prev}} = Z^{\text{curr}}$ **end****return** $n + m - 2\delta$