Article

Timescape: A Simple Spatiotemporal Interpolation Tool

Marco Ciolfi [1,†]*[ID], **Francesca Chiocchini** [1,†][ID], **Rocco Pace** [1][ID], **Giuseppe Russo** [1] and **Marco Lauteri** [1][ID]

[1]   CNR-IRET: Institute of Research on Terrestrial Ecosystems of the National Italian Research Council, via Marconi 2, Porano 05010 Italy.
*   Correspondence: marco.ciolfi@cnr.it – https://www.cnr.it/people/marco.ciolfi
†   These authors contributed equally to this work.

Abstract: We developed a novel approach in the field of spatiotemporal modelling, based on the spatialisation of time: the *Timescape algorithm*. It is especially aimed at sparsely distributed datasets in ecological research, whose spatial and temporal variability is strongly entangled. The algorithm is based on the definition of a spatiotemporal distance that incorporates a causality constraint and that is capable of accommodating the seasonal behaviour of the modelled variable as well. The actual modelling is conducted exploiting any established spatial interpolation technique, substituting the ordinary spatial distance with our *Timescape distance*, thus sorting, from the same input set of observations, those causally related to each estimated value at a given site and time. The notion of causality is expressed topologically and it has to be tuned for each particular case. The Timescape algorithm originates from the field of stable isotopes spatial modelling (isoscapes), but in principle it can be used to model any real scalar random field distribution.

Keywords: Spatiotemporal Modelling; Ecological Modelling; Sparse Data; Minkowskian Geometry; Time Series Analysis; Spatial Statistics; Isoscapes.

MSC: 92-04; 92-08; 92D40; 92F05.

1. Introduction

One of the major issues in ecological modelling is the associated occurrence of spatial and temporal variability of the observations underlying the models [1]. Often one has to merge old-fashioned and modern measurements in a coherent mixture, especially when following the evolution of some phenomenon over time. Concerning both the purely spatial and purely temporal dependence of the variables, we have a rich modelling toolbox at hand, ranging from time series analysis [2,3] to spatial statistics [4–6]. However, when one cannot decide whether to favour spatial or temporal variability, the toolbox offers fewer options, commanding greater statistical intricacy [7–10], see for example the moving average processes [11] or the covariance-based methods [12] implemented in R gstat package [13,14]. In some cases we could have a dynamical model at hand [15] so the modelling is generally conducted via partial differential equations integration, as it is often the case with population dynamics [16]. Other trending approaches include neural networks and deep leaning techniques [17,18]. Furthermore, one is often confronted with the problem of model scaling [19], though the computational limitations are less and less important.

We chose to treat our observations as a random scalar field through standard geostatistical techniques, without introducing an explicit dynamic model or any kind of machine learning. Spatial statistics demands that the distribution of our observations obeys more or less strict constraints, depending on the kind of modelling we would like to perform [4,20]. On the other hand, the distribution of observations in ecology if often sparse, both in time and space, especially when merging new data, collected according to a robust spatial pattern, with old observations, typically collected just at spot sites or along transects: data non-stationarity and a spatially skewed distribution of the observations is often

encountered [21], so the modelling should be conducted, as a matter of principle, also with sub-optimal datasets, maybe at the expense of accuracy.

In order to deal with entangled spatial and temporal variability, we propose a novel spatiotemporal interpolation technique – the *Timescape algorithm* – that estimates the spatiotemporal distribution of a given variable (technically, a *real random field*) from a set of observations. The key idea is borrowed from the machinery of relativistic physics [22,23], which incorporates a notion of causality rooted into its very topological structure, the Minkowski spacetime [24]. Time is treated as new spatial dimension [25], via the multiplication by a constant with the dimensions of a velocity, and we also forbid the instantaneous propagation in space of the field values by filtering the possible sources when estimating the random field.

In a nutshell, the Timescape algorithm works at a topological level, introducing a spatiotemporal distance that mimics the Minkowskian structure, then exploiting any ordinary spatial interpolation tool for actually estimating the field values. Our spatiotemporal distance can be tuned by a couple of parameters, the aforementioned velocity and a causal strictness parameter, plus a form factor related to seasonality. The Timescape distance is exposed in section 2.1 and 2.4 and the actual implementation is discussed at length in [26].

On practical grounds, we sought for a modelling tool that is: 1) not too choosy about the distribution of the input data, 2) capable to accomodate seasonal variability, 3) suitable for both both projected and geographical coordinates, 4) capable to run on affordable hardware in decent times, 5) distributable as open source. The resulting software implementations are a Python module [26] and two Java applications, one for projected [27] and one for geographical coordinates [28]. The software is exposed in section 2.5. A few case studies are discussed in section 3.

2. Methods

A *Timescape model* is a discrete representation of a scalar field on a space-time continuum with a temporal dimension and any number of spatial dimensions (figure 1), i.e. a lattice of spatiotemporal cells (here and henceforth the *target*) whose values are interpolated from a sparse set of observed field values (the *source*). A Timescape can be regarded as a stack of sheets, each of which is an ordinary discrete spatial model, indexed by the time, as well as a sheaf of time series, identified by their spatial location.
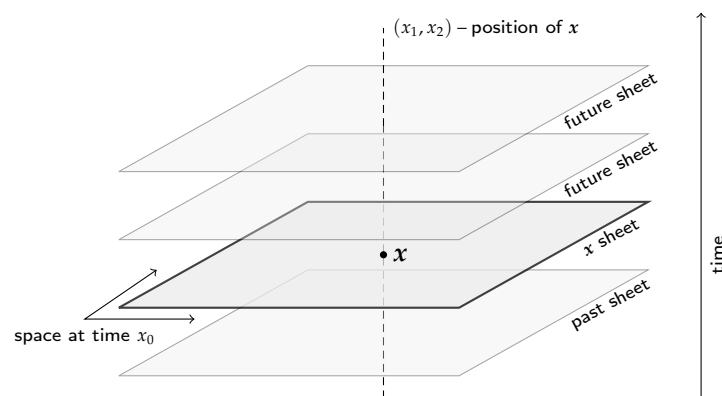
Figure 1. The whole space-time centered about the event x of coordinates (x_0, x_1, x_2): x is located at (x_1, x_2) in a bi-dimensional space at time x_0: $x = (x_0, x_1, x_2)$.

To define a spatio-temporal distance, we mimic the structure of the Minkowskian relativistic metric [22,23,29]. According to the conventions of relativistic physics, we call *event* any space-time element $x = (x_o, x_1, x_2 \ldots x_n)$, where x_0 is the time and $x_1, x_2 \ldots x_n$ are the spatial coordinates.

In the following, we will represent the time and only a single spatial component, regardless the actual number of dimensions (generally two, in both geographical and whatever projected coordinates) – see figure 2.
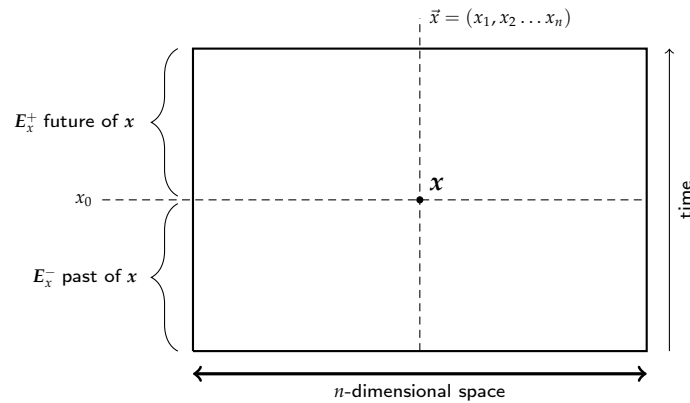
Figure 2. A limited portion of the space-time centred about the event $x = (x_0, x_1, x_2 \ldots x_n)$.

The Minkowskian space-time incorporates a notion of causality due to its topology [24]. In order to implement this geometric realization of causality we must impose an equivalent causal structure on our space-time.

2.1. Space-time distances

Each event x can be split in its time component x_0 and space components $\vec{x} = (x_1, x_2 \ldots x_n)$, i.e. $x = (x_0, \vec{x})$. Now define a *spatial distance* $D_s(x, y)$ between the events x and y using any given ordinary metric function $d : \mathcal{D} \times \mathcal{D} \longrightarrow \mathbb{R}^+$, where $\mathcal{D}$ is the spatial domain of interest (a subset of $\mathbb{R}^n$ or any suitable manifold, e.g. the sphere S^2 of geographical coordinates). A metric $d(\cdot)$ must satisfy the the following conditions: identity: $d(\vec{x}, \vec{y}) = 0 \iff \vec{x} \equiv \vec{y}$, non-negativity: $d(\vec{x}, \vec{y}) \geq 0$, symmetry: $d(\vec{x}, \vec{y}) = d(\vec{y}, \vec{x})$ and subadditivity: $d(\vec{x}, \vec{y}) \leq d(\vec{x}, \vec{z}) + d(\vec{z}, \vec{y}) \ \forall \vec{z} \in \mathcal{D}$, also known as triangle inequality. The latter, in particular, gives the metric its distinctive character of nearness measurement and it is often the key property for the convergence of the interpolation algorithms. We define the spatial distance simply as

$$D_s(x, y) = d(\vec{x}, \vec{y}) \tag{1}$$

The metrics of practical interest in ecological modelling are the ordinary Euclidean distance in $\mathbb{R}^n$ for projected coordinates

$$D_s(x, y) = \sqrt{\sum_{k=1}^{n} \left(x_k - y_k \right)^2} \tag{2}$$

and the geodesic distance in S^2 for geographic coordinates

$$D_s(x, y) = R \arccos \left[\sin x_2 \sin y_2 + \cos x_2 \cos y_2 \cos \left(x_1 - y_1 \right) \right] \tag{3}$$

where R is the radius of the Earth, x_1, y_1 the longitudes and x_2, y_2 the latitudes of the events x and y [30,31].

Since the temporal coordinates live in (a subset of) $\mathbb{R}$, we could naïvely take $|x_0 - y_0|$ as the temporal distance between x and y but the dimensions of $D_s(x, y)$ (length) and $|x_0 - y_0|$ (duration) are not coherent. In order to integrate space and time we introduce the first of the Timescape parameters: a time-to-space *conversion factor* c, i.e. a velocity ($\in \mathbb{R}^+$) associated with each Timescape model.[1] Thus we define the temporal distance

$$D_t(x, y) = c \left| x_0 - y_0 \right| \tag{4}$$

---

[1] This velocity can have or have not a physical interpretation as a propagation/diffusion coefficient, depending on the model. It has to be tuned to an optimal value, see section 2.4.

Now we combine the spatial and temporal distances via the Pythagorean theorem, thus assuring the fulfillment of identity, non-negativity, symmetry and subaddittivity, taking $\sqrt{D_t^2(x,y) + D_s^2(x,y)}$. This function is a well-behaved metric but lacks a causal structure: the point is that the symmetry property commands no distinction between x and y, whatever x_0 and y_0, so we renounce the symmetry constraint, resorting to what is technically a distance in a *pseudo-metric space* [32,33]. The distinctive "measure of closeness" character of a metric is given by its subaddittivity so, respecting the latter constraint, we impose a directionality to our distance, restricting to the subset $x_0 \leq y_0$, thus introducing the asymmetric function $D(x,y)$ from x to y:

$$D(x,y) = \begin{cases} \sqrt{D_t^2(x,y) + D_s^2(x,y)} & \text{if } x_0 \leq y_0 \\ \infty & \text{otherwise} \end{cases} \tag{5}$$

so that if $D(x,y)$ is finite $D(y,x) = \infty$.[2] In a context of spatial interpolation, what (5) is saying is that something happening at x might be a cause of y but y cannot influence x (i.e. no time traveling in the past). We can be more strict requiring that $x_0 < y_0$ is not enough for a source in x to propagate up to y, i.e. we forbid instantaneous actions-at-a-distance, introducing the second Timescape parameter, the *causal parameter k*, positing

$$\Delta(x,y) = \begin{cases} D(x,y) & \text{if } D_s(x,y) \leq k\, D_t(x,y) \\ \infty & \text{otherwise} \end{cases} \tag{6}$$

where $k \geq 0$. Said $E = T \times \mathcal{D}$ the space of all the events, where $T \subseteq \mathbb{R}$ contains all the events' times, the $\Delta(x,y)$ in (6) coincides with $D(x,y)$ in (5) only in a conical subset whose tip is x (figure 3). We call this subset $\mathcal{C}_x^+$, the *future causal cone of x*. It is the locus of the possible outcomes of something happening at x. Formally, $\mathcal{C}_x^+ = \{y \in E \,|\, \Delta(x,y) < \infty\}$.
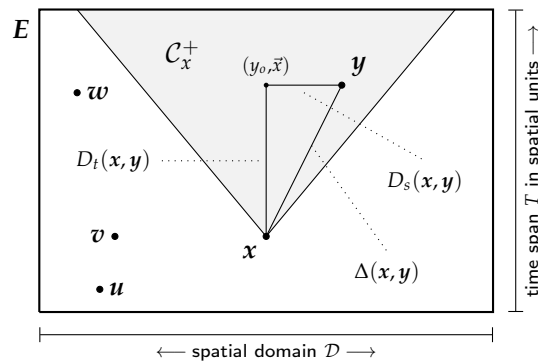
Figure 3. A geometrical interpretation of Δ: $\Delta(x,y)$ is the hypotenuse of the triangle formed by $D_s(x,y)$ and $D_t(x,y)$ if $y \in \mathcal{C}_x^+$. For all the other events $\notin \mathcal{C}_x^+$ the distance is infinte. For example: $\Delta(x,u) = \infty$ because $u_0 < x_0$, $\Delta(x,v) = \infty$ because $v_0 = x_0$ and $\Delta(x,w) = \infty$ because $D_s(x,w) > kD_t(x,w)$ even if $w_0 > x_0$.

---

[2]   The reverse is not true: $D(x,y) = \infty \nRightarrow D(y,x)$ is finite. The distances can be both ∞, e.g. whenever $x_0 = y_0$ and $D_s(x,y) \neq 0$

Exploiting the Heaviside theta function[3] the formal definition of the Timescape distance from x to y can be rephrased as[4]

$$\Delta(x, y) = \frac{\sqrt{D_t^2(x, y) + D_s^2(x, y)}}{\Theta\big[k\, D_t(x, y) - D_s(x, y)\big]} \tag{7}$$

note that (7) is parametrized by c and k: each pair of parameters gives a distinct distance, so a different Timescape model. In section 2.4 we discuss in detail how to choose c and k, maximizing the accuracy of the interpolation.

2.2. Causal Structure: Topology of the events space

Analogously of $\mathcal{C}_x^+$, we introduce the *past causal cone of x*, defined as $\mathcal{C}_x^- = \{y \in E \mid \Delta(y, x) < \infty\}$, something occurring at an event in $\mathcal{C}_x^-$ is a potential cause of x.

The value of k tunes the causal cones width. The bigger k, the larger the cones, the looser the causal constraint. While the smaller k, the thinner the cones, the stricter the constraint (figure 4). As limiting cases, as $k \to 0$, the cones shrink to vertical lines, while as $k \to \infty$, $\mathcal{C}_x^+$ becomes, E_x^+ the whole future of x and $\mathcal{C}_x^-$ becomes E_x^-; $k \to \infty$ corresponds to an infinitely fast propagation of information in E.
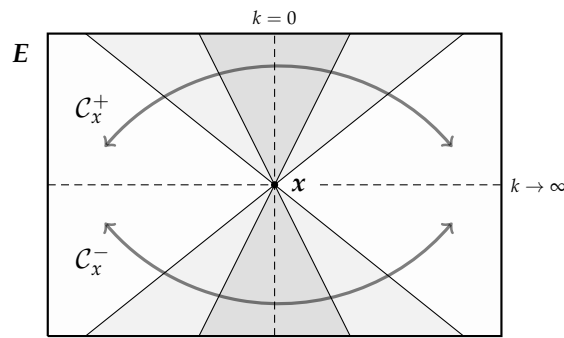
Figure 4. Sheaves of future and past causal cones. x sits at the common tip of the cones. The cones widen as k increases. For $k = 0$ the cones shrink to segments and for $k \to \infty$ the cones coincide to the whole past (E_x^-) and future(E_x^+) of x.

The Timescape algorithm, in a nutshell, consists in filtering topologically the possible causes for each target event from a set of given source events, picking only the source events that fall into the past causal cone of the target event.

Further mathematical details can be found in the manual accompanying the Python Timescape module [26].

2.2.1. Accommodating seasonal variability

In ecological modelling, it is of paramount importance to be able to cope with periodic phenomena. We propose a refinement of the Timescape topological structure that includes seasonality by reshaping the cones by means of a multiplicative positive *form factor* $\psi(t)$ that shrinks and inflates the cone width as $\psi(t) < 1$ or $\psi(t) > 1$, thus modifying the time distance (4) as

$$\widetilde{D}_t(x, y) = c\,|x_0 - y_0|\,\psi\big(|x_0 - y_0|\big) \tag{8}$$

---

[3] The Heaviside function is $\Theta(t) = 1$ if $t \geq 0$ or 0 otherwise, we also posit $\frac{\text{finite}}{0} = \infty$.

[4] It is important to distinguish the starting event (*from x*) from the ending one (*to y*). We cannot simply say "distance *between x and y*" since $\Delta(x, y)$ lacks symmetry; the remnant shadow of symmetry lies in the relation $y \in \mathcal{C}_x^-$ iff $x \in \mathcal{C}_y^+$.
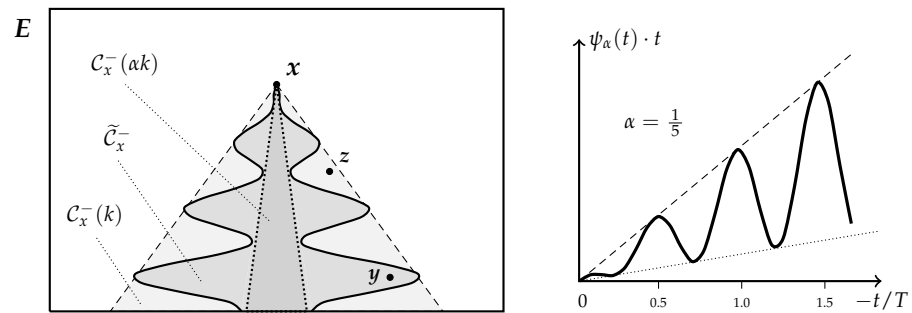
and the Timescape distance (7) accordingly:

$$\widetilde{\Delta}(\boldsymbol{x}, \boldsymbol{y}) \;=\; \frac{\sqrt{\widetilde{D}_t^2(\boldsymbol{x}, \boldsymbol{y}) + D_s^2(\boldsymbol{x}, \boldsymbol{y})}}{\Theta\big[k\, \widetilde{D}_t(\boldsymbol{x}, \boldsymbol{y}) - D_s(\boldsymbol{x}, \boldsymbol{y})\big]} \tag{9}$$

Figure 5 shows an example of periodic shrinking with period T, using the form factor ψ_α, tuned by the parameter α, defined as[5]

$$\psi_\alpha(t) = \alpha + (1 - \alpha)\cos^2\left(\frac{\pi t}{T}\right) \tag{10}$$

with $0 \leq \alpha \leq 1$ (the smaller α, the larger the correction). The resulting "Xmas tree" past cone $\widetilde{\mathcal{C}}_x^-$ shows clearly on- and off-season peaks.


Figure 5. Left: the causal cones $\mathcal{C}_x^-$ and $\widetilde{\mathcal{C}}_x^-$ of x; note that $\widetilde{\Delta}(\boldsymbol{y}, \boldsymbol{x}) < \widetilde{\Delta}(\boldsymbol{z}, \boldsymbol{x}) = \infty$ even if $\Delta(\boldsymbol{y}, \boldsymbol{x}) > \Delta(\boldsymbol{z}, \boldsymbol{x})$. Right: the cross section of $\widetilde{\mathcal{C}}_x^-$, backwards in time, in units of T. The slope of the dashed line is 1 and that of the dotted line is α.

The trick operated by $\psi_\alpha(t)$ in (10) consists in shrinking $\mathcal{C}_x^-(k)$, the regular causal cone of aperture k, down to $\mathcal{C}_x^-(\alpha k)$, i.e. a smaller cone of aperture αk, during the off-season times, thus reducing the number of possible causes of x since $\mathcal{C}_x^-(\alpha k) \subset \widetilde{\mathcal{C}}_x^- \subset \mathcal{C}_x^-(k)$. If $\alpha = 0$ the shrinking is so severe that $\widetilde{\mathcal{C}}_x^-$ becomes a union of saucers connected at single events on the cone axis. The n-dimensional shrunk causal cone $\widetilde{\mathcal{C}}_x^-$ is obtained rotating the $t\,\psi(t)$ function's profile about the horizontal axis (figure 5). In bi-dimensional models it is an ordinary solid of revolution; the rotational symmetry is required by the statistical hypothesis of spatial isotropy, while the widening along time is due to the (finite speed) propagation of the modelled field.

2.3. The Algorithm

A *Timescape model* consists in a set of estimated values of a scalar field Φ, generally arranged as cells of a regular lattice (the *target* $\boldsymbol{T}$). The field is interpolated from a set of known values (the *source* $\boldsymbol{S}$). Each model is characterized by the choice of a spatial distance (the parameters c and k, and the form factor ψ) and the spatial interpolation method. In general, there are no constraints on the distribution of the source events – see figure 6.

---

[5]   The period of $\cos^2$ is half the period of $\cos$, so the cosine argument's normalization factor is π/T.
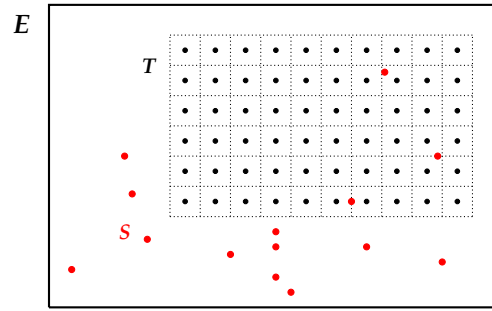
Figure 6. Source (•) and target (•) events. The target events are the centers of a regular lattice of equal cells arranged as an $(n{+}1)$-dimensional box.

Some interpolation methods (e.g. Kriging) require the analysis of the variogram and are also able to evaluate the accuracy of the estimate, some others (e.g. Inverse Distance Weighting) are less demanding but do not provide an accuracy figure. The Timescape algorithm operates a topological filtering of source events before the actual interpolation, so in principle one can choose any tool from the spatial statistical catalogue.

The evaluation of a Timescape variogram deserves a note of caution. First of all, each target event x possesses its own input set S_x, the *causal neighborhood* of x (figure 7), defined as

$$S_x \;=\; S \cap C_x^- \;=\; \big\{ u \in S \,\big|\, \Delta(u,x) < \infty \big\} \tag{11}$$

so, as a matter of principle, each $x \in T$ possesses its own variogram. However, a global variogram can be evaluated by the pairwise comparison of the field values within the source events S.
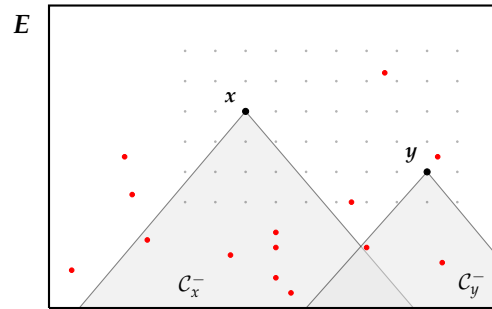
Figure 7. Two target events x and y with their input sets S_x and S_y.

To evaluate the variogram we introduce the set $S_\delta(h)$ of the causally connected pairs of S with a distance within the interval of width δ about h:

$$S_\delta(h) \;=\; \Big\{ (u,w) \in S \times S \;\Big|\; h{-}\tfrac{\delta}{2} \,<\, \Delta(u,w) \,\leq\, h{+}\tfrac{\delta}{2} \Big\} \tag{12}$$

the corresponding variogram value is

$$\gamma(h) \;=\; \frac{1}{\big|S_\delta(h)\big|} \sum_{(u,w)\in S_\delta(h)} \Big(\Phi(u) - \Phi(w) \Big)^2 \tag{13}$$

where $\big|S_\delta(h)\big|$ is the number of elements of $S_\delta(h)$. The bin width δ has to be chosen in order to have no overlapping bins, typically $\delta = \frac{\max\{\Delta\}}{N}$, where N is the optimal number of bins [34] and h is taken at regular intervals $h_n = \big(n{-}\tfrac{1}{2}\big)\delta$ counting $n = 1 \ldots N$. There

is not the usual variogram factor $\frac{1}{2}$ in (13) since the asymmetry of Δ in (12) prevents double-counting.[6]

Generally speaking, for each target event x, the value of the field at x is a weighted average of the field values of the source:

$$\widehat{\Phi}(x) = \sum_{u \in S_x} W\big(u, \Delta(u, x) \dots\big) \, \Phi(u) \tag{14}$$

the actual functional form of the weight W depends on the spatial interpolator of choice, distance being the only common input to all the interpolators. Note that if $S_x = \varnothing$ there is no way to estimate $\widehat{\Phi}(x)$, if this is the case, we attribute a null value to $\widehat{\Phi}(x)$. Other than the field value $\widehat{\Phi}(x)$, one has to calculate the accuracy of the estimate $\widehat{\sigma}(x)$; some interpolators provide such a figure, but some others do not, in the latter case, it is advisable to find the accuracy by jackknifing techniques upon the completion of the interpolation, see section 2.4 for details.
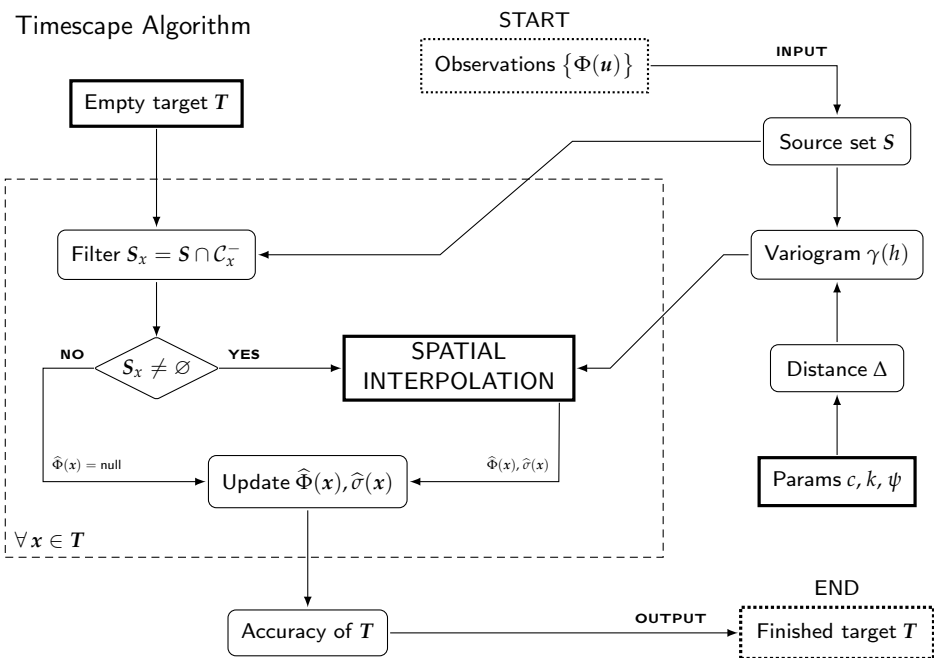
Figure 8. Timescape flowchart. Users must choose a target set T, a spatial interpolator, the distance parameters c and k and form factor ψ (thick boxes). The accuracy can be evaluated by the spatial interpolator itself or a posteriori, depending on the chosen spatial interpolator.

Summing up, the Timescape workflow proceeds as follows (a block flowchart is given in figure 8): firstly, one must create an empty target[7] T, choose a distance function (equation 7 or 9) and a spatial interpolator, then:

---

[6] The intervals in (12) do not overlap but they do not contain 0. This is of no harm since a zero Timescape distance means *at the same place and the same time* and, since Φ has to be single-valued, there cannot exist $u, w \in S$ such that $\Delta(u, w) = 0$ and $\Phi(u) \neq \Phi(w)$.

[7] It is advisable to allocate all the space needed by the finished T before starting the actual interpolation, not to run short of resources during the interpolation. The Python and Java Timescape implementations do so, as a bulk of internal memory (Python) or database records (Java) – see section 2.5.

	for each x in T do	dashed box of figure 8
1	$\mathcal{C}_x^- \leftarrow u \in S \,\vert\, \Delta(u, x) < \infty$	eq.7 eq.9 fig.3
2	$S_x \leftarrow S \cap \mathcal{C}_x^-$	filtering – eq.11 fig.7
3	$\eta(x) \leftarrow \vert S_x \vert$	eq.15
	if $S_x = \varnothing \implies \widehat{\Phi}(x) \leftarrow$ null	
	else $\widehat{\Phi}(x) \leftarrow \sum_{S_x} W(u...)\,\Phi(u)$	estimate – eq.14
4	if **interp.** allows $\widehat{\sigma}(x) \leftarrow \sqrt{\mathrm{var}}$	Kriging…
	else $\widehat{\sigma}(x) \leftarrow$ null	IDW…
5	update x: $\left[\, \widehat{\Phi}(x),\ \widehat{\sigma}(x),\ \eta(x) \,\right]$	

The loop above can be parallelised in a number of ways, since all the events in T are all considered independent of each other. A useful performance figure can be calculated during the interpolation, it is the number $\eta(x)$ of elements in S_x:

$$\eta(x) \;=\; \big\vert\, S_x \,\big\vert \tag{15}$$

the larger $\eta(x)$, the more robust are $\widehat{\Phi}(x)$ and $\widehat{\sigma}(x)$.

Upon the completion of the loop one has to estimate the model's accuracy, if it has not already be done by the interpolator itself. If it is not the case, the accuracy can be estimated by jackknifing the source set [35], removing a few random elements at a time, and repeating such process a number of times N [36], a consistent standard deviation estimate is thus [37]:

$$\widehat{\sigma}(x) \;=\; \sqrt{\frac{1}{N-1}\sum_k \left(\widehat{\Phi}'_k(x) - \widehat{\Phi}(x)\right)^2} \tag{16}$$

where S'_k is the kth replica ($k = 1\ldots N$) of S with some random elements removed, $\widehat{\Phi}(x)$ is the model's field estimate at x and $\widehat{\Phi}'_k(x)$ is its estimate on the subset S'_k. Note that this accuracy estimate is very time-consuming. On practical grounds, (16) can be performed only on a few S'_k sub-sources, since the running time of each subset is comparable with that of the full Timescape model. Note also that one cannot remove too many events from S since it could lead to too many empty S'_k sets, i.e. too many null $\widehat{\Phi}'_k(x)$. Further details on the procedure can be found in Python Timescape module manual [26].

The overall accuracy can be assessed by two figures: a global variance $\sigma^2_{\mathrm{model}}$ and the ratio of null target events over the total target size η_{model}

$$\sigma^2_{\mathrm{model}} \;=\; \frac{1}{\vert T^* \vert}\sum_{x \in T^*} \widehat{\sigma}^2(x) \;, \qquad \eta_{\mathrm{model}} \;=\; \frac{\vert T^* \vert}{\vert T \vert} \tag{17}$$

where $T^* = \left\{ x \in T \,\big\vert\, \widehat{\Phi}(x) \neq \text{null} \right\}$ is the set of not-null elements of T. The η_{model} index, in particular, measures the predictive power of the model. σ_{model} is expressed in the same units of Φ while η is adimensional and $0 \leq \eta \leq 1$.

2.4. Model Tuning

Given the same input set S and output T, the very character of a Timescape model is given by a number of factors: the spatial interpolator, the metric parameters c, k and the form factor ψ. We give some suggestions in order to choose wisely an optimal distance.

2.4.1. Spatial Interpolator

The vast family of Gaussian process regression techniques, (in short, Kriging) provides a wide choice if interpolators, furthermore, such techniques provide an accuracy estimate as a variance figure, so that there is no need to perform the post hoc jackknifing. On the other hand, some old-style deterministic interpolation techniques, like the Inverse Distance Weighting, though lacking the accuracy, are extremely lightweight and fast to perform, so they should be considered, especially when geographic coordinates are involoved, with

a spatial distance like (3).[8] As a rule of thumb, it is advisable to pick the same spatial interpolator one would choose if all the events were located at the same time.

2.4.2. Metric Parameters

c and k affect the metric of E: tuning these values is of paramount importance for a successful interpolation. Other than a trial-and error strategy, one should pick the best performing (c, k) pair. What *best performing* means is a compromise between fast running and accuracy [38]. We recommend a procedure based on residuals minimization by source jackknifing [35]. The idea is based on the cross-estimation of the field values of S with S itself, similar to (16) with S in place of T, for a whole ensemble of (c, k) values, picking the best performing pair $(\bar{c}, \bar{k})$.

Let's define $\widehat{\Phi}_{ck}(\boldsymbol{u})$ the Timescape estimate of $\Phi(\boldsymbol{u})$ –which is known– with parameters (c, k). The estimate is performed on the reduced source set $S \backslash \{\boldsymbol{u}\}$. Mimicking (16) we introduce two indicators, the cumulative residuals $R^2(c, k)$ and the non-null events ratio $N(c, k)$:

$$R^2(c, k) = \frac{1}{|S| - 1} \sum_{\boldsymbol{u} \in S} \left(\widehat{\Phi}_{ck}(\boldsymbol{u}) - \Phi(\boldsymbol{u}) \right)^2 \tag{18}$$

$$N(c, k) = \left| \left\{ \boldsymbol{u} \in S \,\middle|\, \widehat{\Phi}_{ck}(\boldsymbol{u}) \neq \mathsf{null} \right\} \right| \tag{19}$$

the best performing pair $(\bar{c}, \bar{k})$ can be found by minimizing $R^2(c, k)$ and maximizing $N(c, k)$ at the same time. Figure 9 shows an example of R^2 and N surfaces as functions of c and k on a regular lattice. Note that, for a given c, $N(c, k)$ is a decreasing function of k: this is the consequence of the fact that a larger k implies a looser causality (see figure 4).
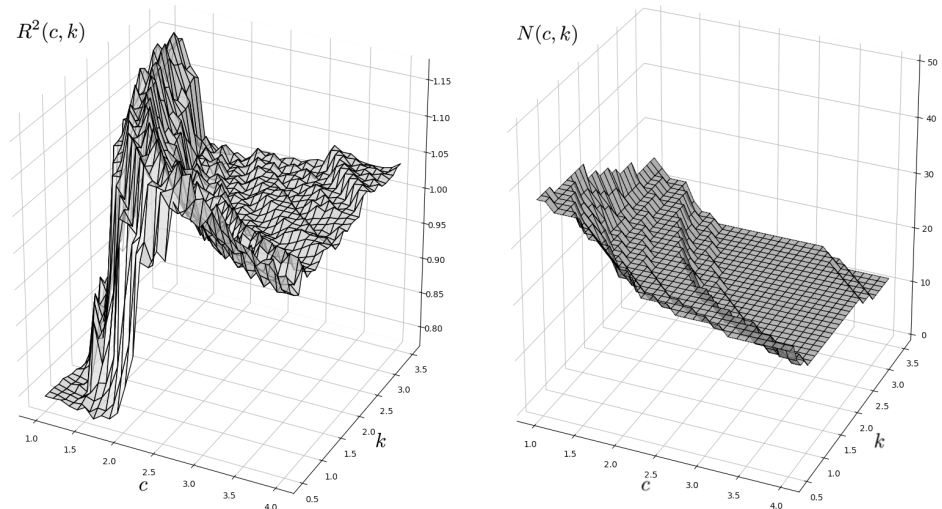
Figure 9. Left: An example of $R^2(c, k)$. Right: $N(c, k)$ in percent units. Plots from a dedicated function of the Python package [26].

As a rule of thumb, any ensemble of (c, k) values should initially include $k_0 = 1$, i.e. treating space and time on equal grounds. As of c, if there is a characteristic velocity in the model (transport, diffusion, etc.) one can play around such a value, otherwise a naïve first guess could be, given $\boldsymbol{u}, \boldsymbol{v} \in S$:

$$c_o = \left[\max_{\boldsymbol{u}} \left(u_0 \right) - \min_{\boldsymbol{u}} \left(u_0 \right) \right]^{-1} \times \max_{\boldsymbol{u}, \boldsymbol{v}} \left[d(\vec{u}, \vec{v}) \right] \tag{20}$$

---

[8] As a matter of principle, one should evaluate the geodesic distances on the Earth as elliptic integrals, while (3) just measures the arc length between the locations $\vec{x}$ and $\vec{y}$.

ensuring a fair coverage of the causal cones.[9]

2.4.3. Form Factor

The topology of the event space-time E is controlled by the form factor ψ. The causal cones are actual cones if $\psi \equiv 1$ or any positive constant, which by the way can be absorbed into k. The form factor is used to modify the shape of the cones and so the topology of E, promoting or depressing the contribution of the events according to their temporal distance. We propose two cases:

Threshold: we can impose a threshold T as a maximum time distance requiring $c\left|y_0 - z_0\right| \leq T$, posing $\psi(t) = \Theta(T - t)$ whose effect is to close the cone $\widetilde{\mathcal{C}}_y^+$. See figure 10 for a comparison with a regular $\mathcal{C}_x^+$, with $\psi(t) = 1$.
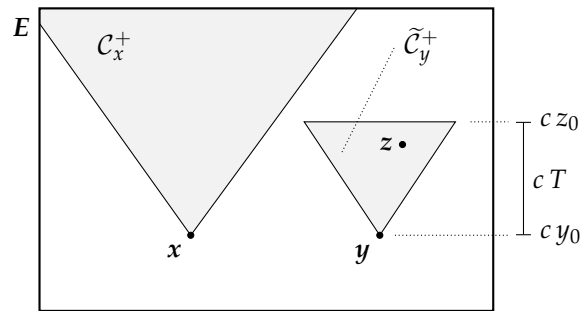
Figure 10. The effect of $\psi(t) = \Theta(T - t)$.

Periodicity: a periodic $\psi(t) \leq 1$ accommodates seasonality into a Timescape model shrinking the causal cone (figure 5), thus filtering out the off-season events.
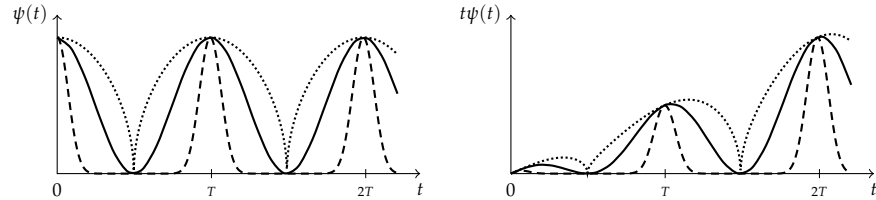
Figure 11. Left: three periodic form factors, solid line $\psi(t) = \cos^2 \omega t$, dashed line (stricter) $\psi(t) = \cos^{20} \omega t$, dotted line (looser) $\psi(t) = \sqrt{|\cos \omega t|}$. Right: the corresponding cross-sections of the causal cones.

Given a period T, a straightforward form factor is $\psi(t) = \cos^2 \omega t$, where $\omega = \pi/T$. If one wants to be more or less selective some variations on the theme are at hand (figure 11). Higher even powers of the cosine have sharper maxima and flat minima, so the cone is more selective. On the other hand, roots of cosine give smoother maxima and sharp minima, corresponding to a looser causal constraint. The cross section of the $\widetilde{\mathcal{C}}^{\pm}$ cones is given by the product function $t\,\psi(t)$.

As a rule of thumb, as a first guess take $\psi(t) = 1$ for non-seasonal field values and $\psi(t) = \cos^2(\frac{\pi}{T} t)$ for seasonal fields modelling, then α-tempering the form with ψ_α in (10), as discussed therein. The larger the area below ψ, the looser the seasonal selectivity.

2.5. Timescape Implementations

Two algorithm implementations have been published: a Python and two Java versions. All the software packages are released under the GNU-GPL license. Other than going deep into the mathematical subtleties of sections 2.1 and 2.2, the manuals accompanying the software distributions treat in detail the practical implementation and describe the code.

---

[9]   For Euclidean spaces (20) makes S base diameter and height equal.

In this paper, we only sketch the key features of the software. Both implementations are limited to two spatial dimensions and provide some statistical analysis tools to explore the source dataset and the finished models.

2.5.1. Python

The Timescape Python module [26] consists in a single module, exposing all the objects, methods and functions needed to perform source data analysis, parameters tuning and the actual model interpolation, plus a bunch of functions dedicated to data plotting and exporting. All the examples in section 3 have been computed with this software.

The usage is command line based, but the functions can be easily integrated into users' own code. The default spatial interpolator exploits the PyKrige module functions [39]; a finished model is a Python object that can be saved in binary format and can be included in a GIS workflow as a multi-layered geotiff image [40], one band per time sheet. The storage is RAM-based, so this is the limiting factor for the number of target cells.

The Python package is designed to be easy configurable, allowing users to code their interpolators. Predefined interpolation methods include Kriging (examples in sections 3.1 and 3.2) and Inverse Distance Weighting (sections 3.3 and 3.4). Other than the Euclidean (2) and Geodesic (3) spatial distances, square and diamond metrics are provided. The allowed form factors are $\psi \equiv 1$ and a periodic $\psi = \cos^2 \frac{\pi t}{T}$ (eq.10, example 3.2). Defining other distances or form factors require a non-trivial intervention on the code.

2.5.2. Java

The Timescape Java implementation consists in two stand-alone applications, one for projected coordinates, the *local* version [27,41], and one for geographical coordinates, the *global* version [28,42]. These are distributed as self-consistent jar executables.

The storage of the Timescape models is based on a Hibernate midlayer [43] that manages the connection with any compatible relational database management system (MySql [44] scripts are included in the distribution). The user interaction is mediated by a Graphical User Interface based on standard swing widgets. The database storage is more flexible, in that each target event is just a record of a table, but there is the need of a dedicated database service.

The Java applications allow a full customization of distance and form factor via a javascript-like scripting language but, since it requires run-time interpretation, the performance is dramatically affected. It is also possible to include ancillary variables along with –or even instead of– the source field values.

The java applications come with a utility for the configuration of the database connection and other parameters. A set of tools is also provided to explore, subset and export the finished models.

3. Results

We present four case studies. They cover a variety of coordinate systems, time scales and peculiarities encountered in ecological geostatistics: projected and geographical coordinates, seasonal effects, uneven distribution and number of observations.

Three examples come from the field of stable isotopes spatial and temporal distribution, also known as isoscapes [45], an increasingly widespread tool in the complex field of spatial ecological analysis [46–48]. The measured isotope ratios are usually expressed as differences with respect to a given reference standard, the so-called δ-notation:

$$\delta^n X \permil \; = \; 1000 \times \frac{R_{\text{sample}} - R_{\text{standard}}}{R_{\text{standard}}} \tag{21}$$

where R is the molar ratio of the heavy ^{n}X stable isotope vs the lighter, most abundant ^{m}X, the factor 1000 rescales the otherwise small numbers to easily readable figures; the deltas are adimensional; for example, δ^{13}C represents the abundance of ^{13}C vs ^{12}C in a given sample. A positive δ^{13}C indicates an enrichment in ^{13}C vs the common ^{12}C with respect

to the standard, while a negative $\delta^{13}C$ denotes a ^{13}C depletion. The interpolated field in such isoscapes is the isotopic abundance (δ^2H, $\delta^{15}N$, $\delta^{18}O$). In one more case study, that demonstrates seasonality, the interpolated field is temperature.

Table 1 resumes the spatial and temporal characteristics of the source events for each model: number of events, geometry, spatial scale and and temporal extension. All the examples are available as extra material, included in the Python software distribution [26].[10]

model	$|S|$	coordinates	$\sim$ scale	time span-resol.
Fungi $\delta^{15}N$	62	Local Euclidean (x, y)	100 m	few days - 1 day
Min temperature	2521	WGS84 UTM33 N	100 km	20 years - 1 month
Olive oil $\delta^{18}O$	275	European Lambert	1000 km	3 years - 1 year
Precipitation δ^2H	1152	Geographical (λ, ϕ)	10^4 km	10 years - 1 year

Table 1. Models' features. $|S|$ is the number of source events.

Table 2 shows the details of the target events set T consistency, the spatial interpolation tool employed (Kriging or Inverse Distance Weighting), the distance function and the causal cone form factor.

model	method	distance	form	target cells $|T|$
Fungi $\delta^{15}N$	Kriging	Euclidean	$\psi \equiv 1$	$64 \times 128 \times 128 = \mathbf{1\,048\,576}$
Min temperature	Kriging	Euclidean	eq.10 $\alpha = 0$	$240 \times 25 \times 32 = \mathbf{192\,000}$
Olive oil $\delta^{18}O$	IDW	Euclidean	$\psi \equiv 1$	$7 \times 74 \times 99 = \mathbf{51\,282}$
Precipitation δ^2H	IDW	Geodesic	$\psi \equiv 1$	$20 \times 120 \times 50 = \mathbf{120\,000}$

Table 2. Models' parameters. The target cells are ordered by time $\times$ horizontal $\times$ vertical.

In the following we briefly describe each case.

3.1. Fungi $\delta^{15}N$

This dataset comes from a study about host trees – mycorrhiza relationship conducted via geostatistical methods exploiting carbon and nitrogen stable isotopes abundances [51]. The model's scalar field represents the hypogeous fungis' ascomata $\delta^{15}N$ (*Tuber aestivum* Vittad. – edible black truffles). The complex tree-fungi relations include the nitrogen transport from fungi to the host tree, in particular, the $\delta^{15}N$ difference, or *fractionation*, is a clue of tree-fungi symbiosis [52–54], with a significant difference in terms of $\delta^{15}N$, resulting in a fungal ^{15}N enrichment with respect to the host [55,56]. From a spatio-temporal point of view, the collection took place within 64 days of continuous fungi-tree interactions, so it was impossible to treat spatial and temporal variability on different grounds. This study led to the development of the Timescape algorithm.

The source consists of 62 events, repeated measures at the same sites, skew-distributed both in space and time. The target is a stack of 64 time sheets counting 128×128 spatial cells each (about one million cells).

Figure 12 illustrates the typical Timescape behaviour, time increasing from bottom to top: as a new observation comes in, there is a sort of perturbation in the field that pushes its value towards that of the following ones. This perturbation will eventually consolidate or fade according to the closest subsequent source events. The round shapes are due to the choice of Euclidean metric (2) as the spatial distance.

When treating cases like this one, it is important to check the accuracy of the interpolation, so it is advisable to choose a statistical interpolator like Kriginig, which is capable

---

[10] For the sake of reference, all the examples are accompanied by R's SpatialPointsDataFrame objects [14,49] with the corresponding projection's CRC objects [50].
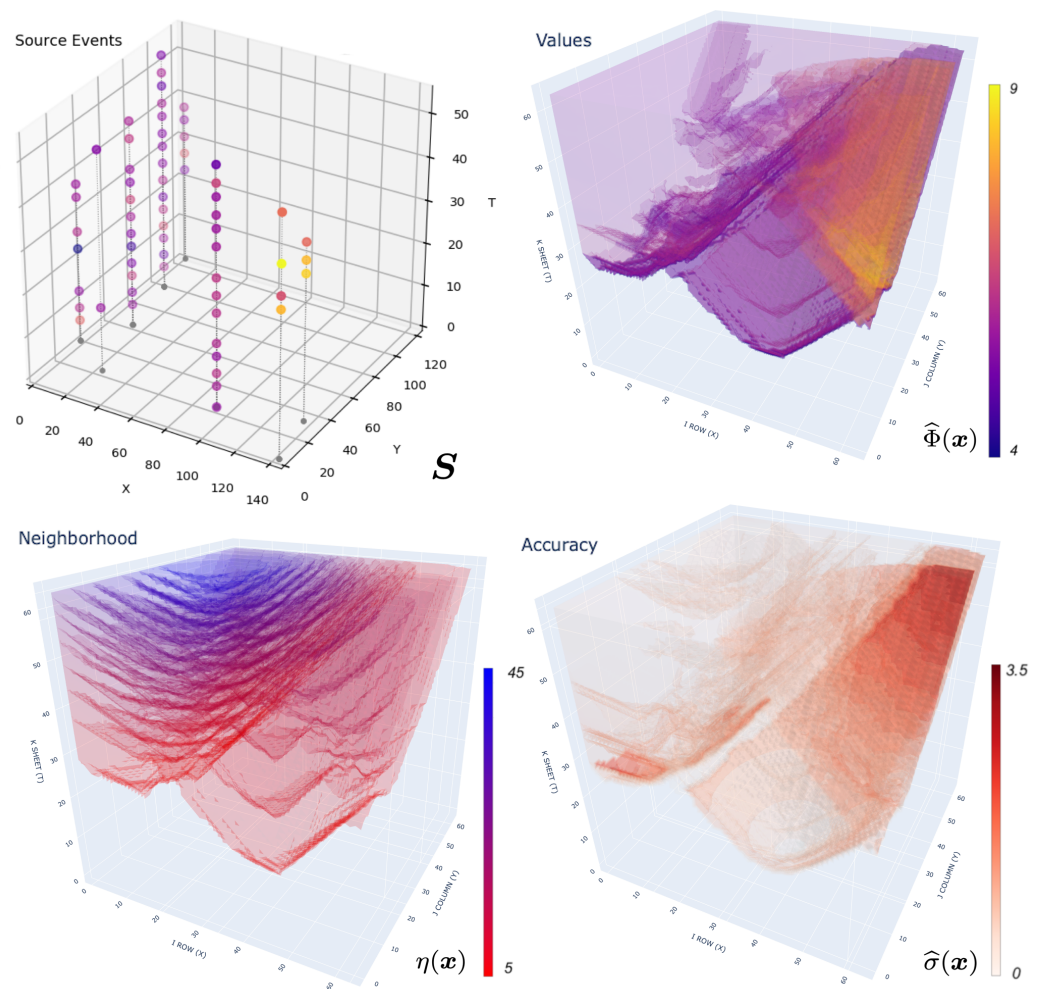
Figure 12. Top left: source events S. Top right: target field values $\widehat{\Phi}(x)$. Bottom left: $\eta(x)$ i.e. events in the causal neighborhood S_x. Bottom right: accuracy $\widehat{\sigma}(x)$. The source coordinates are the actual ones, the target coordinates are cell indexes.

to calculate a target event's variance estimate. Such variance could be relatively high but it is noteworthy that a time-unaware geostatistical interpolation would be impossible to perform altogether. As it appears from figure 12, accuracy stabilizes about the value 1 as time goes on; the upper right corner is meaningless since, spatially, it lies out of the convex hull of the source events, it would be a non-informative area with any ordinary geostatistical tool.

3.2. Lowest temperatures

This case study shows the ability of the Timescape algorithm to cope with seasonal variability. The source dataset consists in the monthly lowest temperatures recording of Umbria region, in central Italy [57,58].

The temperature records span 20 years with monthly time resolution (1/12 yr), their spatial resolution increases over time as more and more stations have been added in the two decades considered. The interpolation has been conducted with a periodic causal cone ($T = 1$ yr) using universal Kriging. The great number of source events, more than 2000, added further computational load: in fact, the throughput is as low as 30 target events per second, see table 3 in section 3.5.

The source events are shown in figure 13. Note the temporal gaps in the measurements. Also, new stations started logging data over time, resulting in denser events from year 1990
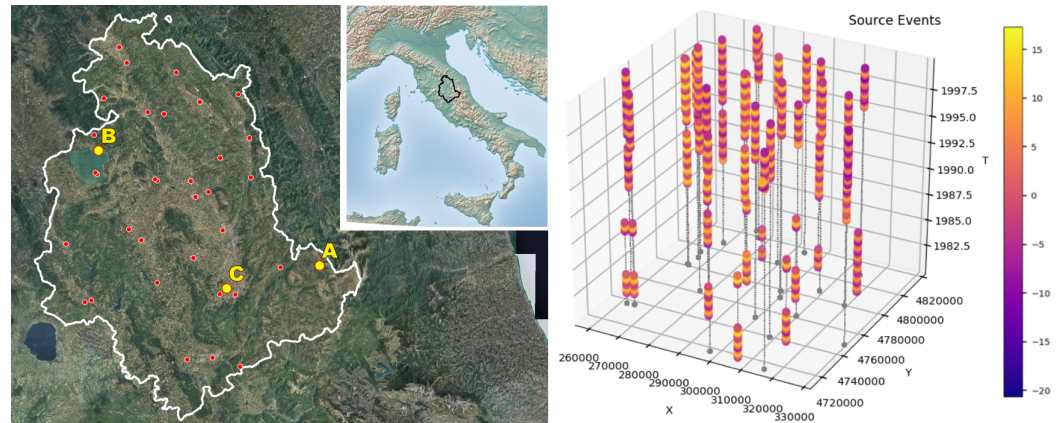
Figure 13. Left: The weather stations network (red dots) in Umbria, the labels are referred to the time series of figure 14. Right: spatio-temporal distribution of the source events. Base map and boundaries from NaturalEarthData [59] and Geoportale Nazionale [60].

on, while around 1980 the collection is sparse in both space and time. This is the typical distribution skewness of surveys based on a growing measurement network.
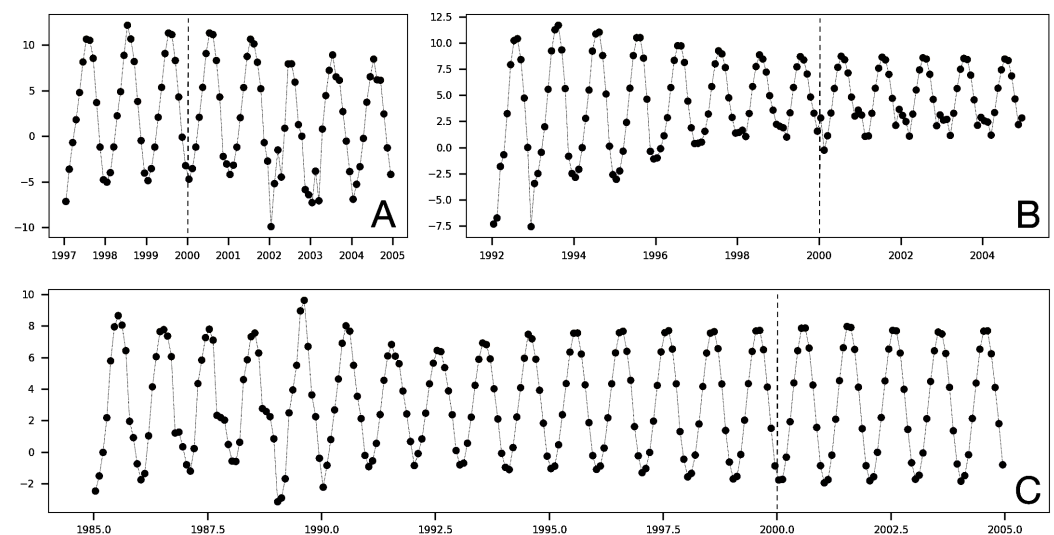

Figure 14. Time series extracted at the locations marked in figure 13. Time is in years and field values in °C. The dashed lines mark the beginning of extrapolation.

The seasonal character of the output is apparent in the time series shown in figure 14: the upper time is 2005 (five years later the source data), so in fact we are forecasting the values: the ups and downs on the left of the dashed line are interpolated, while on the right the values are extrapolated, their period remains constant (one year, of course) and the damping effect is negligible. Seasonal variability is cumbersome for Kriging [61]; in our case, the Timescape algorithm filtered out the off-season source values, thus reproducing the correct periodic behavior. The plots shown in figure 14 have different starting times, corresponding to the earliest meaningful time (i.e. the first non-null $\widehat{\Phi}(x)$) for the the given site. The time resolution of the series is one month. The B series of figure 14 is initially influenced by outlying low values, due to the influence of a distant mountain station, which eventually becomes irrelevant, as closer measuring stations started logging data.

3.3. Extra Virgin Olive oil $\delta^{18}O$

This dataset comes from a study on the geographic traceability of Italian Extra Virgin Olive Oil through isoscape analysis [62] based on the spatial distribution of Carbon [63–

65] and Oxygen [66–68] isotopic composition of precipitation and source water [69]. The δ^{18}O isoscapes, in particular, are related to the elevation and to the distance from the sea and correlate with the year's precipitation [66,67]; the original results in [62] have been obtained by standard spatial statistical procedures. The Timescape interpolation of the same data gives results comparable with [62] for the first year only (figure 15, left), while the subsequent years (figure 15, centre and right) do not match well with the source events. Note in particular how the extreme values are averaged out in 2010 and 2011: this example shows clearly that the Timescape algorithm is not a universal tool, in this case, each year's data are self-contained and independent of each other, so mixing three years of observations is not correct.
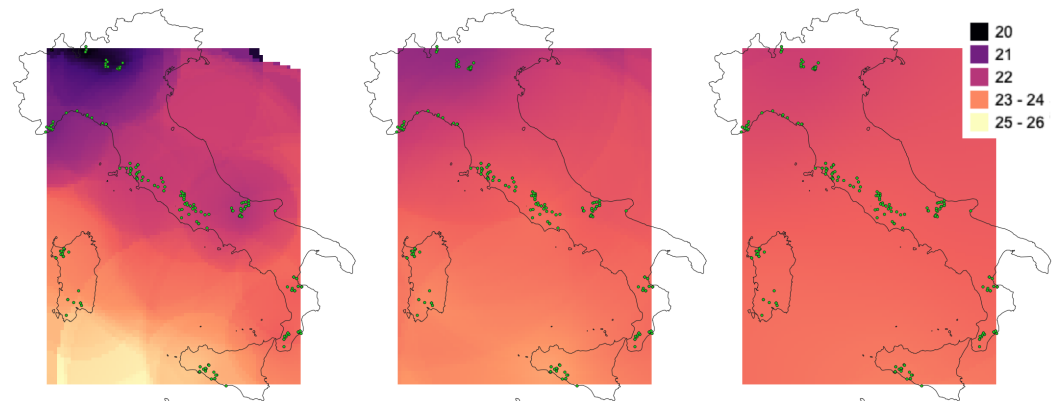
Figure 15. 2009, 2010 and 2011 δ^{18}O isoscapes as sheets of the model. The source events' locations are shown in green. Country boundaries from NaturalEarthData [59], plots obtained in Qgis [70] from exported geotiff images [40].

The computation per se, however, is straightforward: the high voxels throughput (more than one thousand per second, see table 3) is due to the relatively scarce number of source events and, mostly, to the computational speed of the IDW. In this case the weight function (14) is simply $W(\boldsymbol{u}, \boldsymbol{x}) = \Delta(\boldsymbol{u}, \boldsymbol{x})^{-1}$. It is worth noting that the same model interpolation by Kriging gave rise to a significant number of errors, about 35% of the target events, in spite of the causal cones relatively large width ($k = 5$). This is due to the fact that each year's δ^{18}O integrates that particular year's water intake of the olive tree, and it is unrelated to other years' values, resulting in different values with the same spatial coordinates, which is particularly disturbing for Kriging [4] – the high number of target events' computational errors is an alarm bell signalling a possible inconsistency of the source events' values or a badly chosen interpolation tool, as it is in this case.

3.4. Precipitation $\delta^2 H$

This example is based on the renowned Global Network of Isotopes in Precipitation (GNIP) dataset [71,72] from the International Atomic Energy Agency [73]. The collection of isotopic abundance data started in the early 1960s and the stations network has grown dramatically over time. For historical and technical reasons, the spatial coverage is uneven (figure 16), with the earlier, denser coverage in central Europe.

The interpolation algorithm chosen is a smooth version of the Inverse Distance Weighting method, with a weight function

$$W(\boldsymbol{u}, \boldsymbol{x}) \;=\; \frac{1}{\Delta(\boldsymbol{u}, \boldsymbol{x})^2 + m^2} \tag{22}$$

where m^2 is a positive *inertial term*, that prevents the overweighting of the closest source events. In our case $m^2 = 2$. Moreover, a maximum number of ten neighbors has been imposed. Nonetheless, the computation is CPU-intensive since the geodesic metric (3) involves slow trigonometric calculations. Limiting the causal neighborhood requires S_x to

be ordered by distance, then trimmed retaining only the closest events to x; this is based on the heuristic observation that only the closest elements contribute significantly to the weight functions. Both Python and Java Timescape implementations allow this kind of trimming.
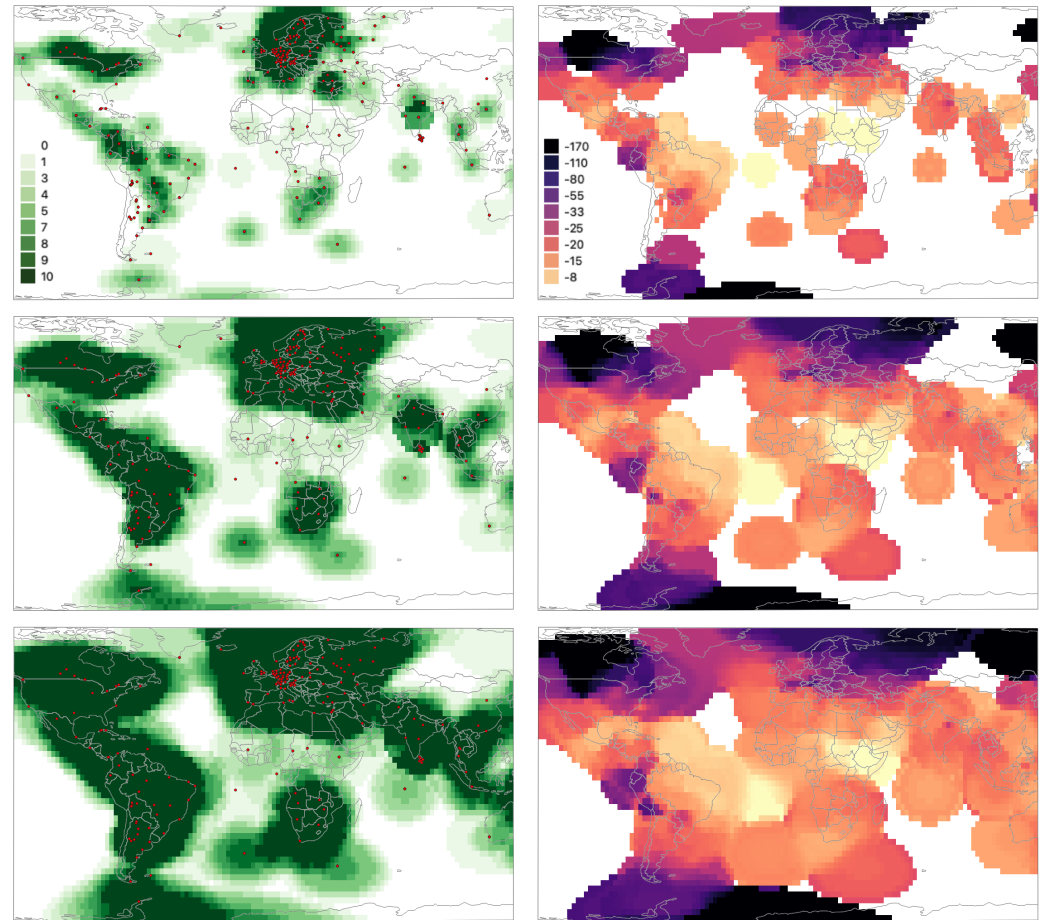
Figure 16. Three sheets of the Timescape model, time increasing top to bottom (year 1981, 1985 and 1989). Left: Causal neighborhood $\eta(x)$, source events' sites are shown as red dots. Right: field values $\widehat{\Phi}(x)$, note the typical bull's eye shapes due to the IDW algorithm.

This example shows how the interpolation results are affected by the addition of new sampling sites. The dates covered are form year 1975 to 1984 (included). This is of course a very crude model, based only on the GNIP δ^2H observations, further improvements could benefit of a regionalised regression based on climatic ancillary data [67,74]. An acknowledged source of Hydrogen isoscapes is the web-based application IsoMAP [75], that also provides insightful information about precipitation hydrogen and oxygen isoscapes in general [68].

3.5. Performance Analysis

The interpolation throughput has been estimated in terms of evaluated target events per second, typical values range from a few tens to more than one thousand, depending on three factors: the number of source events $|S|$, the spatial interpolator (see the method column in table 2, Kriging is slower than IDW), the distance function (geodesic is much slower than Euclidean) and the causal cone form factor (periodic is slower than straight). The cone shape and width also influences the number of null target events. Table 3 resumes the modelling performance of the Python Timescape version on three quite standard hw/sw configurations, no claiming exhaustiveness:

- **configuration 1**: laptop – Intel i7 @ 2.60 GHz, 4 cores, 16 GByte RAM, SSD storage, Python 3.8 on Windows 10 OS.
- **configuration 2**: laptop – Intel i7 @ 2.30 GHz, 4 cores, 8 GByte RAM, SSD storage, Python 3.7 on Ubuntu Linux 19.10 OS.
- **configuration 3**: virtual machine – Oracle Virtualbox on MacPro host, Intel Xeon-E5 @ 3.50 GHz – 4 dedicated cores, 16 GByte reserved RAM out of 64, SSD storage, Python 3.8 on Xubuntu Linux 20.04-LTS OS.

model	η_{model}	configuration 1		configuration 2		configuration 3	
		time	evt/s	time	evt/s	time	evt/s
Fungi δ^{15}**N**	55%	87'32"	**200**	125'2"	**140**	75'33"	**231**
Min temperature	73%	115'49"	**28**	118'6"	**27**	78'20"	**41**
Olive oil δ^{18}**O**	81%	39"	**1315**	47"	**1091**	35"	**1478**
Precipitation δ^2**H**	62%	15'42"	**157**	16'4"	**124**	11'21"	**176**

Table 3. Models' performances, measured in target events per second. η_{model} (17) is expressed in percent, running times in minutes' seconds".

The lowest throughput is associated with the temperature model (section 3.2). In fact, this model associates a large source set, a periodic form factor and a Kriging spatial interpolator, each factor adding its computational load.

A fair estimate of the python Timescape throughput can be set roughly about a few hundreds events per second with a standard hardware configuration. A conservative estimate of 100 evt/s is a good starting point for a new model.

We give no figures for the Java applications since their time performance depends essentially on the database connection speed.

4. Discussion

The outcome of a Timescape model is based on three elements: the source set distribution, the spatiotemporal geometry (i.e. the shape of the causal cone and the parameters c and k) and the chosen spatial interplation method. Picking a spatial interpolator is a matter of personal taste and computational performance; anyway it is worth stressing that the Timescape distance is independent of (logically, it precedes the) spatial interpolation itself, so two models based on the same source, same topology, but different interpolators shouldn't be much different than two ordinary spatial models are.

As for the scale of modelling, using geographical coordinates adds further complexity, due to the geodesic distance (3). If it is possible,[11] it is advisable to divide the whole target set into patches which are tractable with projected coordinates, then merge the results, as it could be done with any ordinary spatial interpolator.[12]

We can explore further topological diversions modifying the structure of distance; section 4.1 introduces a few suggestions. Altering the definition of the causal cones has a deep impact on causality, so the results can be unexpected even if the mathematical procedure itself is sound. A possible scenario could be inferring the past from the future (properly, *retrodiction*), exchanging the role of past end future cones $\mathcal{C}_x^-$ and $\mathcal{C}_x^+$.

Accuracy assessment is the most delicate phase of Timescape modelling. On the one hand, statistical interpolators provide an accuracy figure $\hat{\sigma}(x)$ along with the estimated field $\hat{\Phi}(x)$ at running time, but it is also possible to apply jackknifing (16) on the finished target for every interpolator. The Python Timescape module [26] offers a set of functions (see figure 9) for an initial guess of model's accuracy.

---

[11] The patches must be statistically "rich enough" and the Timescape distance parameters have to be the same across the patches. A necessary hypothesis is that far apart patches are independent of each other.

[12] Geometrically, one builds an *atlas* of compatible *charts*. Special care is required since the spherical coordinates are periodic and singular at the poles.

4.1. Timescape Topology Modification

The distance (7) can be modified in a more general way than than just applying a form factor that is only a function of time (9). As a matter of principle, the past causal cone $\mathcal{C}_x^-$ of x can have any shape, provided that it is a subset of the past of x: $\mathcal{C}_x^- \subseteq E_x^-$, where $E_x^- = \{ y \in E \mid y_0 \le x_0 \}$ – see figure 2. As a limiting case, $\mathcal{C}_x^-$ could be the whole x's past E_x^-. Any such cone shape would retain the property $\Delta(x, y) < \infty \implies \Delta(y, x) = \infty$ but not necessarily $y \in \mathcal{C}_x^- \iff x \in \mathcal{C}_y^+$.
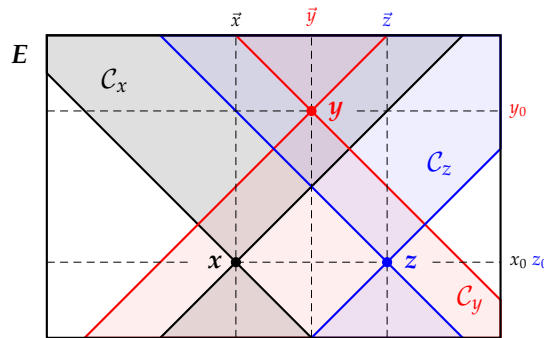
Figure 17. Symmetric double causal cones. Note that $\Delta(x, y) < \infty$ and $\Delta(y, z) < \infty$, nonetheless $\Delta(x, z) = \infty$ because $x_0 = z_0 \wedge \vec{x} \neq \vec{z} \implies x \notin \mathcal{C}_z$.

Another modification of (7) consists in including retrodiction; just glue $\mathcal{C}_x^-$ and $\mathcal{C}_x^+$ by the common tip x as a double cone $\mathcal{C}_x$. This restores symmetry in the metric function ($x \in \mathcal{C}_y \iff y \in \mathcal{C}_x$), but at the price of subadditivity, that only holds if, given $x, y, z \in E$, the three events are mutually causally connected, i.e. $x \in \mathcal{C}_y$, $y \in \mathcal{C}_z$ and $z \in \mathcal{C}_x$ (the converse is guaranteed by symmetry). Otherwise symmetry is violated, as clarified in figure 17. Furthermore, allowing retrodiction requires a factor one-half in front of variogram (13) to prevent double-counting.

Giving up causality altogether, the topological filtering can still be performed, with any x-neighbourhood in lieu of $\mathcal{C}_x$ double cone, but hardly it would be of any use in ecological modelling. It is worth mentioning that taking the whole E as $\mathcal{C}_x$ for all x, though respecting both symmetry ad subadditivity, just means performing a standard spatial interpolation in one more dimension, with no topological filtering at all: this is not a Timescape model any more.

Other than geometrically, the distance (7) can be modified according to a set of external factors, i.e. exploiting the knowledge of one or more ancillary variables. For example, the form factor-modified time distance $\widetilde{D}_t$ (8) could be a function of some weather-related variables other than the time

$$\widetilde{D}_t(x, y) = c \, |x_0 - y_0| \, \psi(x_0, y_0) \, \Psi(a_1(x) \ldots a_n(x), a_1(y) \ldots a_n(y)) \tag{23}$$

where the a_k are the observations of ancillary variables at the x and y events. Note that $\psi(x_0, y_0)$ is more general than $\psi(|x_0 - y_0|)$ in (8).

The Java applications [27,28] allow the mentioned modifications of distance function and the inclusion of ancillary variables, via a scripting language. The Python module [26] can only use symmetrical cones in the event's past, allowing at most $\mathcal{C}_x^- = E_x^-$ as a limiting case, letting $k \to \infty$. Practically, a large enough k suffices, say $k = \max D_s(u, x) / \min D_t(u, x)$ where $u \in S$, $x \in T$ belong to the source and target sets, respectively.

4.2. Timescape vs Minkowskian Geometry

Since the seminal work of Minkowski [76] the causal structure of relativistic space-time has been encoded in the so-called *light cone*. Causally connected events lie within each other's light cone, while no signal can reach two events outside the cones. Furthermore,

the speed of light c acts as the obvious conversion factor between spatial and temporal coordinates. So the Minkowskian space-time is naturally equipped with a topological structure that incorporates causality straightforwardly.

The Timescape geometry purposely introduces a light cone like structure in the ordinary, non-relativistic, space-time. The role of the speed of light is shared between our c and k parameters, and the causal connection has to be introduced somehow forcefully, but the resulting topology is almost identical.

As far as topology is concerned, the Timescape and Minkowskian spaces act similarly, but the metric behaviour is completely different. Using our symbols, the Timescape D_T and Minkowskian D_M squared distances are[13]

$$D_T^2(\boldsymbol{x}, \boldsymbol{y}) = D_t^2(\boldsymbol{x}, \boldsymbol{y}) + D_s^2(\boldsymbol{x}, \boldsymbol{y})$$
$$D_M^2(\boldsymbol{x}, \boldsymbol{y}) = D_t^2(\boldsymbol{x}, \boldsymbol{y}) - D_s^2(\boldsymbol{x}, \boldsymbol{y})$$

(24)

The minus sign of D_M in (24) encodes the light cone structure. Furthermore, D_M is not even a distance since D_M^2 is not positive definite and, even when $D_M^2 \geq 0$, it does not behave as a distance; on the boundary of the light cone $D_t = D_s \implies D_M = 0$, which describes perfectly the light in the relativistic arena, but it is far from what an ordinary distance is expected to do. On the other hand, the plus of D_T in (24) ensures subadditivity, i.e. D_T restricted to the causal cone is a proper distance, in fact, it is just a Pythagorean sum of squared sides (figure 3). Summarizing, our Timescape distance mimics the Minkowskian topological structure, not its metric. The trick is operated by the $\Theta(kD_t - D_s)$ in the denominator of (7) that makes Δ singular.[14]

4.3. A universal tool?

The Timescape algorithm can be exploited, in principle, with any set of observed variables varying in space and time. The Timescape algorithm should not be intended as a universal tool, though. In figure 18 we sketch a decision tree to help decide whether it should be chosen or not, depending on the distribution of input observations.
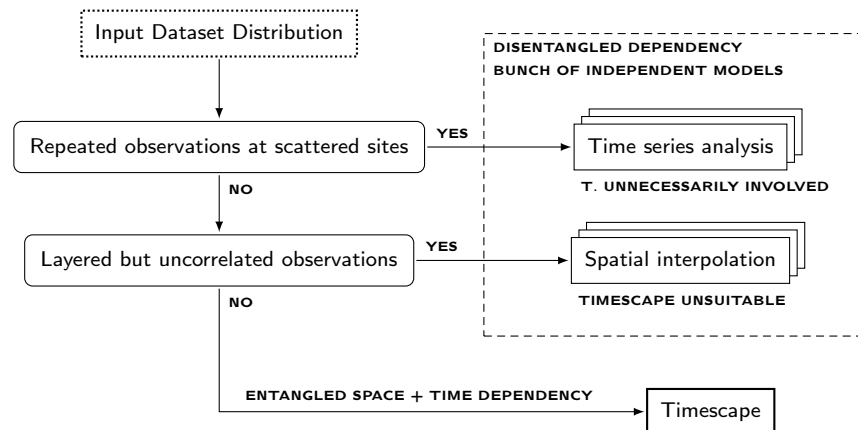

Figure 18. Observations-driven decision tree.

In particular, it has not to be chosen whenever there is a clear temporal or spatial pattern in the source data that allows the disentanglement of spatial and temporal variability by subsetting the input set – this is the case, for example, of the Olive oil δ^{18}O case study,

---

[13] The sign of $D_M^2(\boldsymbol{x}, \boldsymbol{y})$ defines, for each event $\boldsymbol{x}$, three different space-time regions: the so-called *time-like* one, corresponding to the interior of our causal cones, the *space-like* region of causally unreachable events, and their common boundary, the locus of *light-like events* $\boldsymbol{y}$ whose $D_M(\boldsymbol{x}, \boldsymbol{y}) = 0$, reserved for light (or other massless particles). This is not the kind of space one expects to find in ecological modelling: $D_T(\boldsymbol{x}, \boldsymbol{y}) = 0$ means that $\boldsymbol{x}$ an $\boldsymbol{y}$ coincide in space and time, unlike $D_M(\boldsymbol{x}, \boldsymbol{y}) = 0$ that includes all the light-like events' pairs.

[14] The topological correspondence is: Minkowski space-like interval $\rightarrow$ Timescape $\Delta = \infty$, time-like $\rightarrow \Delta < \infty$, light-like $\rightarrow \Delta < \infty$ too. There is nothing special to the events laying on the boundary of the causal cones.

section 3.3. On the other hand, when the input dataset can be clearly partitioned as a bunch of time series, the algorithm can be applied, but it is unnecessarily complicated and time-consuming; in this case, the result is comparable with ordinary interpolated one-dimensional time series.

5. Conclusions

The Timescape algorithm is a modelling tool aimed at the interpolation of spatiotemporal field distributions from recorded observations presenting an entangled variability in space and time. The spatial and temporal records of the modelled field are embedded into a spacetime inspired by the causal topology of the Minkowskian space, the well-known arena of relativistic physics. This is achieved applying the distance function (7) for filtering the causally connected events before submitting the actual field estimate to an ordinary spatial interpolator. The choice of the interpolator is up to the user, as well as a couple of parameters that tune the distance to fit the particular input data. Seasonality, often encountered in ecological modelling, can be dealt with modifying the distance function (7) by means of a form factor (9).

The Timescape algorithm has been already implemented as a Python module and as two Java stand-alone GUI-based applications. The software is distributed under the GNU-GPL open license. The software is meant to be used with georeferenced input data and can be included in a GIS workflow, eventually exporting geotiff files.

Our modelling tool can cope with seasonality by playing with the shape of the causal cone, without subsetting the input set season-by-season. Moreover, the algorithm can be applied with a variety of interpolation methods. Ideally, any spatial interpolator would do, the choice being driven by the input data and the user's taste.

Author Contributions: Conceptualization MC, FC, GR and ML; Methodology MC and FC; Formal analysis MC, FC and RP; Data curation MC, FC, RP and GR, Visualization MC and GR, Software development MC and FC; Software testing RP and GR, Funding acquisition ML and MC, Manuscript curation MC, FC, RP, GR and ML. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: Timescape is published under the GNU-GPL3 license. The source code and the case studies' data are available to download. The Python package along with the sample data is available at [26] and the Java implementations are available at [27] and [28].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Evans, M.R. Modelling ecological systems in a changing world. *Philosophical Transactions of the Royal Society B: Biological Sciences* **2012**, *367*, 181–190. doi:10.1098/rstb.2011.0172.
2. J. Beran. *Mathematical Foundations of Time Series Analysis*; Springer International Publishing, 2017. doi:10.1007/978-3-319-74380-6.
3. R. H. Shumway, D. S. Stoffer. *Time Series Analysis and Its Applications*; Springer International Publishing, 2017. doi:10.1007/978-3-319-52452-8.
4. N. Cressie. *Statistics for Spatial Data*; John Wiley & Sons, Inc., 1993. doi:10.1002/9781119115151.
5. Chilès, J.P.; Delfiner, P. *Geostatistics*; John Wiley & Sons, Inc., 2012. doi:10.1002/9781118136188.
6. Delicado, P.; Giraldo, R.; Comas, C.; Mateu, J. Statistics for spatial functional data: some recent contributions. *Environmetrics* **2009**, *21*, 224–239. doi:10.1002/env.1003.
7. N. Cressie, C. K. Wikle. *Statistics for Spatio-Temporal Data*; John Wiley & Sons, Inc., 2015.
8. C. K. Wikle, A. Zammit-Mangion, N. Cressie. *Spatio-Temporal Statistics with R*; Chapman and Hall/CRC, 2019.
9. Kyriakidis, P.C.; Journel, A.G. Geostatistical space-time models: a review. *Mathematical Geology* **1999**, *31*, 651–684. doi:10.1023/a:1007528426688.
10. Nhu D. Le, James V. Zidek. *Statistical Analysis of Environmental Space-Time Processes*; Springer Series in Statistics, 2006.

11. Epperson, B.K. Spatial and space–time correlations in ecological models. *Ecological Modelling* **2000**, *132*, 63–76. doi:10.1016/s0304-3800(00)00305-7.

12. Pebesma, E. spacetime: Spatio-Temporal Data in R. *Journal of Statistical Software* **2012**, *51*, 1–30.

13. Graeler, B.; Pebesma, E.; Heuvelink, G. Spatio-Temporal Interpolation using gstat. *The R Journal* **2016**, *8*, 204–218. doi:10.32614/RJ-2016-014.

14. R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, 2018.

15. Chen, Q.; Han, R.; Ye, F.; Li, W. Spatio-temporal ecological models. *Ecological Informatics* **2011**, *6*, 37–43. doi:10.1016/j.ecoinf.2010.07.006.

16. Holmes, E.E.; Lewis, M.A.; Banks, J.E.; Veit, R.R. Partial Differential Equations in Ecology: Spatial Interactions and Population Dynamics. *Ecology* **1994**, *75*, 17–29. doi:10.2307/1939378.

17. Christin, S.; Hervet, É.; Lecomte, N. Applications for deep learning in ecology. *Methods in Ecology and Evolution* **2019**, *10*, 1632–1644. doi:10.1111/2041-210x.13256.

18. Brodrick, P.G.; Davies, A.B.; Asner, G.P. Uncovering Ecological Patterns with Convolutional Neural Networks. *Trends in Ecology & Evolution* **2019**, *34*, 734–745. doi:10.1016/j.tree.2019.03.006.

19. Fritsch, M.; Lischke, H.; Meyer, K.M. Scaling methods in ecological modelling. *Methods in Ecology and Evolution* **2020**, *11*, 1368–1378. doi:10.1111/2041-210x.13466.

20. Wang, J.F.; Stein, A.; Gao, B.B.; Ge, Y. A review of spatial sampling. *Spatial Statistics* **2012**, *2*, 1–14. doi:10.1016/j.spasta.2012.08.001.

21. Jun, M.; Stein, M.L. Nonstationary covariance models for global data. *The Annals of Applied Statistics* **2008**, *2*, 1271–1289. doi:10.1214/08-aoas183.

22. T. Frankel. *The Geometry of Physics*; Cambridge University Press, 2009. doi:10.1017/cbo9781139061377.

23. J. Jost. *Geometry and Physics*; Springer Berlin Heidelberg, 2009. doi:10.1007/978-3-642-00541-1.

24. A. Ashtekar and V. Petkov., Ed. *Springer Handbook of Spacetime*; Springer Berlin Heidelberg, 2014. doi:10.1007/978-3-642-41992-8.

25. Basseville, M. Distance measures for signal processing and pattern recognition. *Signal Processing* **1989**, *18*, 349–369. doi:10.1016/0165-1684(89)90079-0.

26. M. Ciolfi. Timescape Python module, GNU-GPL 3 license. https://github.com/ciolfis/timescape, accessed on 01.12.2021.

27. M. Ciolfi. TimescapeLocal Java application, GNU-GPL 3 license. https://sourceforge.net/projects/timescapelocal, accessed on 01.12.2021.

28. M. Ciolfi. TimescapeGlobal Java application, GNU-GPL 3 license. https://sourceforge.net/projects/timescapeglobal, accessed on 01.12.2021.

29. J. Jost. *Riemannian Geometry and Geometric Analysis*; Springer Berlin Heidelberg, 2011. doi:10.1007/978-3-642-21298-7.

30. Banerjee, S. On Geodetic Distance Computations in Spatial Modeling. *Biometrics* **2005**, *61*, 617–625. doi:10.1111/j.1541-0420.2005.00320.x.

31. E. E. Mamajek et al.. IAU 2015 Resolution B3 on Recommended Nominal Conversion Constants for Selected Solar and Planetary Properties, 2015, [arXiv:astro-ph.SR/1510.07674].

32. Ambartzumian, R.V. A note on pseudo-metrics on the plane. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* **1976**, *37*, 145–155. doi:10.1007/bf00536777.

33. Schroeder, V. Quasi-metric and metric spaces. *Conformal Geometry and Dynamics of the American Mathematical Society* **2006**, *10*, 355–361. doi:10.1090/s1088-4173-06-00155-x.

34. Scott, D.W. On optimal and data-based histograms. *Biometrika* **1979**, *66*, 605–610. doi:10.1093/biomet/66.3.605.

35. Efron, B.; Stein, C. The Jackknife Estimate of Variance. *The Annals of Statistics* **1981**, *9*, 586–596. doi:10.1214/aos/1176345462.

36. Knight, K. *Mathematical Statistics*; Chapman & Hall CRC, 2000.

37. McIntosh, A. The Jackknife Estimation Method, 2016, [1606.00497].

38. Bennett, N.D.; Croke, B.F.; Guariso, G.; Guillaume, J.H.; Hamilton, S.H.; Jakeman, A.J.; Marsili-Libelli, S.; Newham, L.T.; Norton, J.P.; Perrin, C.; et al. Characterising performance of environmental models. *Environmental Modelling & Software* **2013**, *40*, 1–20. doi:10.1016/j.envsoft.2012.09.011.

39. PyKrige developers. PyKrige, 2D and 3D Python Kriging implementation. https://pykrige.readthedocs.io, accessed on 01.12.2021.

40. Open Geospatial Consortium. OGC GeoTIFF Standard. https://www.opengeospatial.org/standards/geotiff, accessed on 01.12.2021.

41. M. Ciolfi, F. Chiocchini, M. Mattioni, M. Lauteri. Timescape Local spacetime interpolation tool: projected coordinates java standalone application. *Smart eLab* **2017**, *10*, 20–39. doi:10.30441/smart-elab.v10i0.201.

42. M. Ciolfi, F. Chiocchini, M. Mattioni, M. Lauteri. Timescape Global spacetime interpolation tool: geographic coordinates java standalone application. *Smart eLab* **2018**, *11*, 1–51. doi:10.30441/smart-elab.v11i0.202.

43. Hibernate object relational mapping. http://hibernate.org, accessed on 01.12.2021.

44. Oracle. MySQL Developer Zone. https://dev.mysql.com, accessed on 01.12.2021.

45. West, J.B.; Bowen, G.J.; Dawson, T.E.; Tu, K.P., Eds. *Isoscapes*; Springer Netherlands, 2010. doi:10.1007/978-90-481-3354-3.

46. Wagner, H.H.; Fortin, M.J. Spatial Analysis of Landscapes: Concepts and Statistics. *Ecology* **2005**, *86*, 1975–1987. doi:10.1890/04-0914.

47. Rossi, R.E.; Mulla, D.J.; Journel, A.G.; Franz, E.H. Geostatistical Tools for Modeling and Interpreting Ecological Spatial Dependence. *Ecological Monographs* **1992**, *62*, 277–314. doi:10.2307/2937096.

48. West, J.B.; Bowen, G.J.; Cerling, T.E.; Ehleringer, J.R. Stable isotopes as one of nature's ecological recorders. *Trends in Ecology & Evolution* **2006**, *21*, 408–414. doi:10.1016/j.tree.2006.04.002.

49. R.S. Bivand, E. Pebesma, V. Gomez-Rubio. *Applied spatial data analysis with R*; Springer, New York, 2013. doi:10.1007/978-1-4614-7618-4.

50. Pebesma, E.; Bivand, R. Classes and Methods for Spatial Data in R. *R News* **2005**, *5*.

51. M. Ciolfi, G. Piovesan (supervisor). Spatiotemporal analysis and modeling of ecological processes at ecosystem, landscape and bioregion scale. PhD dissertation, Università della Tuscia, Viterbo, 2016.

52. Hogberg, P.; Hogberg, M.N.; Quist, M.E.; Ekblad, A.; Nasholm, T. Nitrogen isotope fractionation during nitrogen uptake by ectomycorrhizal and non-mycorrhizal Pinus sylvestris. *New Phytologist* **1999**, *142*, 569–576. doi:10.1046/j.1469-8137.1999.00404.x.

53. Hobbie, E.A.; Colpaert, J.V. Nitrogen availability and colonization by mycorrhizal fungi correlate with nitrogen isotope patterns in plants. *New Phytologist* **2003**, *157*, 115–126. doi:10.1046/j.1469-8137.2003.00657.x.

54. Dawson, T.E.; Mambelli, S.; Plamboeck, A.H.; Templer, P.H.; Tu, K.P. Stable Isotopes in Plant Ecology. *Annual Review of Ecology and Systematics* **2002**, *33*, 507–559. doi:10.1146/annurev.ecolsys.33.020602.095451.

55. Henn, M.R.; Chapela, I.H. Ecophysiology of 13C and 15N isotopic fractionation in forest fungi and the roots of the saprotrophic-mycorrhizal divide. *Oecologia* **2001**, *128*, 480–487. doi:10.1007/s004420100680.

56. Govindarajulu, M.; Pfeffer, P.E.; Jin, H.; Abubaker, J.; Douds, D.D.; Allen, J.W.; Bücking, H.; Lammers, P.J.; Shachar-Hill, Y. Nitrogen transfer in the arbuscular mycorrhizal symbiosis. *Nature* **2005**, *435*, 819–823. doi:10.1038/nature03610.

57. Umbria Region Hydrographic Service. https://servizioidrografico.regione.umbria.it, accessed on 01.12.2021.

58. Annali Regione Umbria. Annals of the Umbria Hydrographic Service. https://annali.regione.umbria.it, accessed on 01.12.2021.

59. Natural Earth public domain vector and raster maps. https://www.naturalearthdata.com, accessed on 01.12.2021.

60. Geoportale Nazionale of the Italian Ministry for the Environment. http://www.pcn.minambiente.it, accessed on 01.12.2021.

61. Giraldo, R.; Delicado, P.; Mateu, J. Continuous Time-Varying Kriging for Spatial Prediction of Functional Data: An Environmental Application. *Journal of Agricultural, Biological, and Environmental Statistics* **2010**, *15*, 66–82. doi:10.1007/s13253-009-0012-z.

62. F. Chiocchini, S. Portarena, M. Ciolfi, E. Brugnoli, M. Lauteri. Isoscapes of carbon and oxygen stable isotope compositions in tracing authenticity and geographical origin of Italian extra-virgin olive oils. *Food chemistry* **2016**, *202*, 291–301. doi:10.1016/j.foodchem.2016.01.146.

63. Brugnoli, E.; Farquhar, G.D. Photosynthetic Fractionation of Carbon Isotopes. In *Photosynthesis*; Springer Netherlands, 2000; pp. 399–434. doi:10.1007/0-306-48137-5_17.

64. Portarena, S.; Gavrichkova, O.; Lauteri, M.; Brugnoli, E. Authentication and traceability of Italian extra-virgin olive oils by means of stable isotopes techniques. *Food Chemistry* **2014**, *164*, 12–16. doi:10.1016/j.foodchem.2014.04.115.

65. Camin, F.; Larcher, R.; Nicolini, G.; Bontempo, L.; Bertoldi, D.; Perini, M.; Schlicht, C.; Schellenberg, A.; Thomas, F.; Heinrich, K.; et al. Isotopic and Elemental Data for Tracing the Origin of European Olive Oils. *Journal of Agricultural and Food Chemistry* **2010**, *58*, 570–577. doi:10.1021/jf902814s.

66. Longinelli, A.; Selmo, E. Isotopic composition of precipitation in Italy: a first overall map. *Journal of Hydrology* **2003**, *270*, 75–88. doi:10.1016/s0022-1694(02)00281-0.

67. *Isotopic Composition of Precipitation in the Mediterranean Basin in Relation to Air Circulation Patterns and Climate*; Number 1453 in TECDOC Series, International Atomic Energy Agency: Vienna, 2005.

68. Bowen, G.J.; Revenaugh, J. Interpolating the isotopic composition of modern meteoric precipitation. *Water Resources Research* **2003**, *39*. doi:10.1029/2003wr002086.

69. Camin, F.; Larcher, R.; Perini, M.; Bontempo, L.; Bertoldi, D.; Gagliano, G.; Nicolini, G.; Versini, G. Characterisation of authentic Italian extra-virgin olive oils by stable isotope ratios of C, O and H and mineral composition. *Food Chemistry* **2010**, *118*, 901–909. doi:10.1016/j.foodchem.2008.04.059.

70. QGIS Development Team. QGIS Geographic Information System. Open Source Geospatial Foundation Project. http://qgis.osgeo.org, accessed on 01.12.2021.

71. The Global Network of Isotopes in Precipitation of the International Atomic Energy Agency. https://www.iaea.org/services/networks/gnip, accessed on 01.12.2021.

72. *Statistical Treatment of Data on Environmental Isotopes in Precipitation*; Number 331 in Technical Reports Series, International Atomic Energy Agency: Vienna, 1992.

73. IAEA. International Atomic Energy Agency. https://www.iaea.org, accessed on 01.12.2021.

74. Terzer, S.; Wassenaar, L.I.; Araguás-Araguás, L.J.; Aggarwal, P.K. Global isoscapes for $\delta^{18}O$ and δ^2H in precipitation: improved prediction using regionalized climatic regression models. *Hydrology and Earth System Sciences* **2013**, *17*, 4713–4728. doi:10.5194/hess-17-4713-2013.

75. IsoMAP – Isoscapes modeling, analysis and prediction. http://isomap.org, accessed on 01.12.2021.

76. Minkowski, H. Das Relativitätsprinzip. *Annalen der Physik* **1915**, *352*, 927–938. doi:10.1002/andp.19153521505.