Received: 16 September 2017 | Accepted: 31 December 2017

DOI: 10.1111/nrm.12158

WILEY Natural Resource Modeling

Algorithmic geolocation of harvest in hand-picked agriculture

Nitin Srivastava[1] | Peter Maneykowski[1] | Richard B. Sowers[1,2]

[1]Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana–Champaign, Urbana, IL (Email: nsrivas3@illinois.edu; maneyko2@illinois.edu)

[2]Department of Mathematics, Rapid IoT (RIoT) Laboratory, University of Illinois at Urbana–Champaign, Urbana, IL

Correspondence
Richard B. Sowers, Department of Industrial and Enterprise Systems Engineering, Department of Mathematics, Rapid IoT (RIoT) Laboratory, University of Illinois at Urbana–Champaign, Urbana, IL 61801.
Email: r-sowers@illinois.edu

Funding information
University of Illinois

Abstract

Precision agriculture significantly depends on measuring yield; this allows feedback to optimize various decisions. While spatially granular yield mapping is readily available in machine-harvested row crops, it is more difficult in hand-picked row crops. We study here a data set collected during harvesting of strawberries; using smartphones, we collected Global Positioning System (GPS) logs of individual harvesters. Using recent advances in feature identification, we are able to algorithmically decompose the path into individual excursions into the field to harvest the berries. This lays the groundwork for yield mapping.

To further develop this area, we recommend that Resource Managers

- Pursue wider scale trials of geolocated harvest data collection of hand-picked crops.
- Join this geolocated harvest data with data from other aspects of field operations.
- Join this geolocated harvest data with output measurements like quality and quantity.

KEYWORDS

hand-picked crops, precision agriculture, yield information

1 | INTRODUCTION

One of the developing success stories in the application of “big data” is *precision agriculture*: using very granular data to analyze, optimize, and predict the production and processing of various agricultural products. World population is estimated to increase to 9.7 billion in 2050, from a current count

https://doi.org/10.1111/nrm.12158

of 7.3 billion United Nations (2015). Increasingly ubiquitous sensing abilities (e.g., *in situ* nitrogen and soil sensors, and aerial imagery from drone and satellite) allow precise understanding of the state of the field. For row crops (wheat, corn, soy, etc.), large mechanized farming equipment provides natural platforms from which to gather very precise data on harvesting; advances in connectivity have made that data immediately available. The scientific understanding of agriculture has, as a result, seen significant advances.

The promise of precision data is that it will enable better *decision making*. Food production is in fact a very complicated industry. It depends heavily on long-term weather behavior, functional machinery, prices of the food products themselves, prices of raw materials, energy, and water, and, in many cases, on human involvement.

Our interest here is in algorithmically studying and identifying several operational aspects in *hand-picked* high-value specialty crops: strawberries, certain types of grapes and apples, and others. While many crops are harvested by machine, a number of crops crucially depend on human discernment. U.S. production of hand-picked crops is often in the billions of dollars. These crops can have very high value (multiple times the value of row crops). Since a mechanized platform (e.g., harvesters) is often currently unavailable for these crops (in part due to the need for human discretion in identifying fruit of the correct ripeness), some aspects of precision agriculture need to be rethought.[1]

Labor is often the largest expense in the whole process[2] for these crops. The central role of human behavior leads to a number of relatively novel issues. Variability of harvesting is much greater than in machine-harvested crops. Efficiency and speed vary from harvester to harvester and from moment to moment. See Gemtos, Fountas, Tagarakis, and Liakos (2013) for a study of precision agriculture in hand-picked fruits in Greece, where Normalized difference vegetation index (NDVI) and yield variability are compared. Our effort on labor analytics will hopefully complement the data analytics of field fluctuations; fluctuations of field and labor should be considered jointly.

This paper focusses on reverse-engineering precise behavior in a field based on *GPS tracks*. There is an increasing interest in using passively obtained GPS information as a source of information. GPS data can be treated as a “fingerprint”; see Rossi, Walker, and Musolesi (2015). Similarly, GPS data can be used to identify modes of mobility Ellis et al. (2014), Zheng, Li, Chen, Xie, and Ma (2008). In the case of hand-picked agriculture, we believe that this type of information from GPS data can be used as the foundation of a yield map.

Mathematically, our challenge is to identify several prespecified patterns in GPS tracks. Simply put, harvesting occurs when a harvester makes an excursion into a field. We would like to algorithmically identify this behavior, particularly in real data, which is discrete and noisy. We also are faced with heterogeneity in harvesting speed and behavior; some harvesters move faster than others, and there are several typical types of excursions.

Starting with data collected from a strawberry farm in Southern California, we develop an L^2 minimization problem, which captures our pattern-matching problem. We adapt some recent work of Liu and Li (2016), which seeks sparse solutions of pattern-matching problems. We then follow this with some *ad hoc* techniques for cleaning and simplifying the resulting suggested patterns.

2 | SETUP AND PROBLEM

Our efforts here focus on data collected during harvest at Crisalida farms, in Oxnard, California, on February 19, 2016. Four harvesters were equipped with smartphones, which recorded GPS coordiantes at 5-second intervals. The latitude and longitude tracks are in Figure 2. Looking at the latitude and longitude, one can visually see lunch breaks, harvesting, and weeding activity (where the harvesters walk the whole length of the field).

A focal point of our efforts is to algorithmically interpret geolocation data. In this case, we want to disambiguate several behaviors; we want to quantify human labor as much as possible. Since human labor is extremely costly, this may be useful in helping to optimize it.

- How should labor be deployed? How should a crew of laborers be optimally configured? Labor is typically deployed in a *crew* of about seven to 10 workers. This crew is supervised, as a group, by a field boss, and mobile collection equipment (see Section 4) is placed relative to the crew. Harvesters are often paid piece rate (i.e., amount harvested) and on an hourly basis. One of the workers in a crew is usually at least slightly faster than the others, and typically he or she sets the pace for the rest of the crew. However, if the fastest worker then ends up being idle, this is likely suboptimal. Moreover, there are in fact various types of nonharvesting but necessary behavior–weeding, stacking of crates, etc. Better identification of these different tasks may allow one to better assess pay schedules. With realtime data, labor might be better and dynamically redeployed.
- How can farm production be optimized around labor? Given the high cost of labor, how can equipment placement and timing be optimized? One of the challenges is to identify as much as possible simply through geospatial tracks.
- For highly perishable crops, cooling is an important part of maintaining product value. Can one automatically and dynamically schedule transportation to cooling facilities based on data (cf. Manuel & Sowers, 2017)?

There are also other operational concerns surrounding labor. By law, harvesters must be given lunch breaks and warmup time. Can data analytics provide sufficient verification of good labor practices?

3 | METHODS

Our work reflects data captured at Crisalida farms in Oxnard, California, on February 19, 2016. Four harvesters carried smartphones with an app, which logged position and timestamp roughly every 5 seconds. The smartphones were of various age and precision (Nexus 4's to Maven ZTE's).

Figure 1 gives an approximate trajectory of a typical harvester's path. The rows in the field were in this case oriented north–south at this time, so harvesting within a row corresponded to north–south (latitudinal) motion, and motion across the rows accounted for east–west (longitudinal) motion. Starting at point 1 in Figure 1, the harvester makes an excursion into the field along a row to point 2. At point 2, the harvester starts harvesting as he or she returns to point 3 along the edge of the field (following the red solid line from point 2 to point 3 in Figure 1; red lines in Figure 1 denote harvest, while yellow dashed lines denote nonharvest motion). The harvester off-loads (and submits for inspection) his or her harvest at a collection station located along the edge of the field near point 3. The harvester then moves to a new row (point 4) and starts another excursion into the field. The fruit along this new row may be a bit thin, so the harvest makes an excursion into a row in the adjacent field (point 6 to point 7) and continues harvesting (point 7 to point 8) to fill up his or her cart before again off-loading.

Several things are not apparent from Figure 1. The collection area moves along the edge of the field from time to time; the harvester may have to move along the edge of the field to go to the collection area. Second, during harvest activities (the solid red lines in Figure 1), the harvester moves a bit more

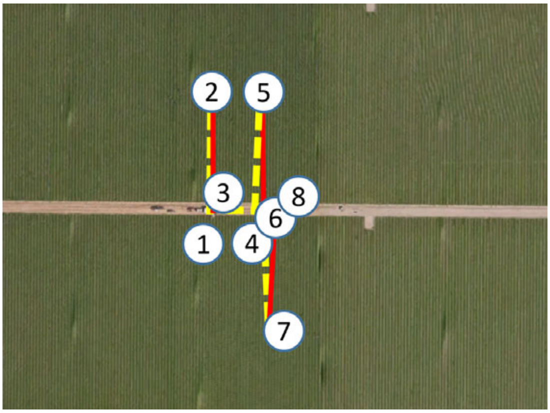

FIGURE 1 Track

slowly. When the harvest is particularly thick, the harvester may make more than one excursion into the same row. Each harvester in a sense “owns” his or her rows; as he or she harvests, the harvester culls the berries (i.e., picks diseased or malformed berries and throws them on the ground) in preparation for the next harvest several days later. Figure 1 shows one harvester's path. Harvesting is typically carried out by a team of seven to 10 harvesters; the rows between the red lines 2 to 3 and 5 to 6 “belong” to a different harvester.

The recorded latitude and longitude data are in Figure 2. Let's look specifically at the first harvester, Harvester 0, whose latitude and longitude traces are in Figure 3. By interpreting these figures, can we reverse-engineer what is going on, in terms like our explanation of Figure 1? The amount of data is initially fairly small, and insufficient for a machine-learning approach. This paper focusses on a feature-identification approach. In Figure 3, the latitude trace shows a number of excursions into the field, while the longitude trace shows a gradual motion across the field as row after row is harvested. Figure 1 captures the main activities, but there may be minor variations. Furthermore, the actual data behind Figure 3 includes GPS noise. Nevertheless, we would like to identify excursions into the field and tag them as harvest activity. Having these data will allow us to, over the long term, help identify yield as a function of soil and microlocal ground conditions.

4 | RESULTS AND DATA

The starting point of the analysis (of the behavior of Harvester 0) is a collection of geolocated tracks taken at observation instants $\mathcal{T} = \{t_1, t_2 \dots t_N\}$. These times need not be equally spaced; in some cases battery or connectivity problems may cause a record to be lost. At time t_n, a record (x_n, y_n) records the longitude (the x value) and latitude (the y value). Since the rows were in north–south orientation; the excursion into the field can be identified solely by looking at the latitude (y) coordinate. Between such excursions, the harvester stays on an east–west edge-of-field path along the side of the field; this edge-of-field path lies at constant latitude

$$\hat{y}^* \approx 34.0001. \tag{1}$$

A mobile collection station was stationed on the edge-of-field path; at various times, this collection station was repositioned to be as close as possible to where the harvesters were currently working.

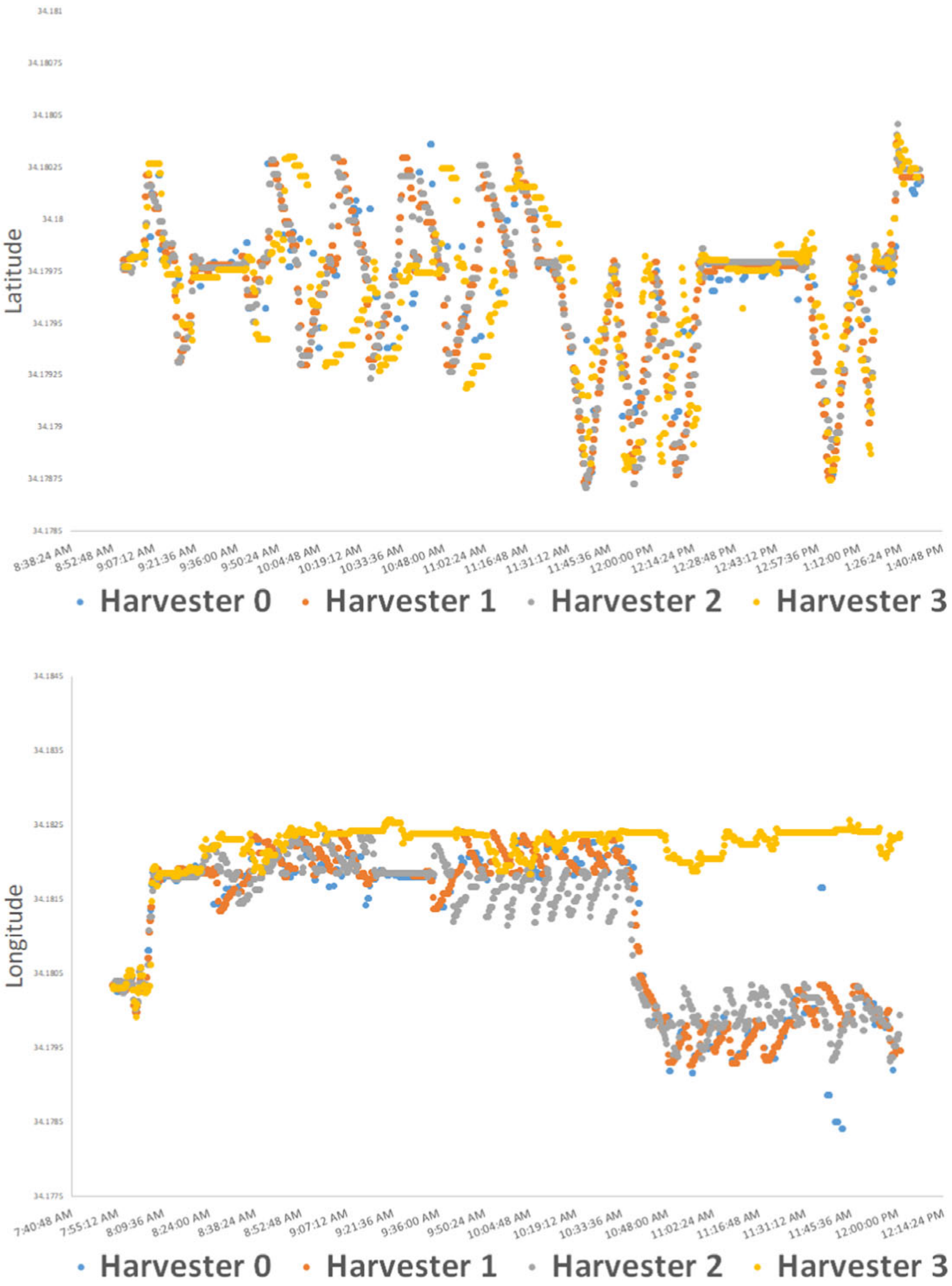

FIGURE 2 Latitude (top) and longitude (bottom)

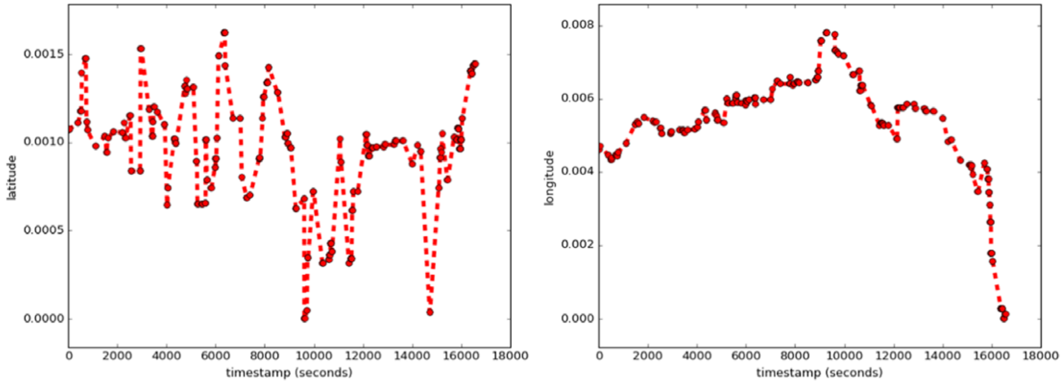

FIGURE 3 Latitude (left) and longitude (right) for Harvester 0

Next, define a reference "excursion." The data are discrete, and sometimes the harvester "runs" from the edge-of-field to the region where he/she is harvesting. They may harvest in a particular area, and then quickly return back to the collection area at the edge of the field. Consider a reference "Box" function

$$B^\circ(t) \overset{\text{def}}{=} \chi_{[0,1]}(t) = \begin{cases} 1 & \text{if } 0 \leq t \leq 1 \\ 0 & \text{else} \end{cases}$$

and then shift and scale it; for (h, T, w) in

$$\mathcal{U} \overset{\text{def}}{=} \mathbb{R} \times \mathbb{R} \times (0, \infty), \tag{2}$$

define

$$B_{h,T,w}(t) \overset{\text{def}}{=} hB^\circ\left(\frac{t-T}{w}\right), \qquad t \in \mathbb{R}. \tag{3}$$

The set $\mathcal{U}$ represents the universe of parameters for all possible "excursions."

As an example, look at the first 50 records of data; see Figure 4. There is an excursion of about $w_1 \overset{\text{def}}{=} 200$ seconds and height $h_1 \overset{\text{def}}{=} 0.003$ degrees latitude starting about $T_1 \overset{\text{def}}{=} 515$ seconds after the first data point. There is a second excursion of about $w_2 \overset{\text{def}}{=} 400$ seconds and height $h_2 \overset{\text{def}}{=} -0.0026$ degrees latitude (an excursion of negative height corresponds to an excursion below $\hat{y}^*$) starting about $T_2 \overset{\text{def}}{=} 2550$ seconds after the first data point. Roughly, the track followed by these two excursions can be written as the function

$$t \mapsto B_{h_1,T_1,w_1}(t) + B_{h_2,T_2,w_2}(t) + \hat{y}^*. \tag{4}$$

See Figure 4 (which contains several more excursions). A bit more conveniently, define

$$\boxed{\hbar \overset{\text{def}}{=} 0.0002} \tag{5}$$

as a reference excursion height; (4) can then be rewritten as

$$t \mapsto \alpha_1 B_{\hbar,T_1,w_1}(t) + \alpha_2 B_{\hbar,T_2,w_2}(t) + \hat{y}^*, \tag{6}$$

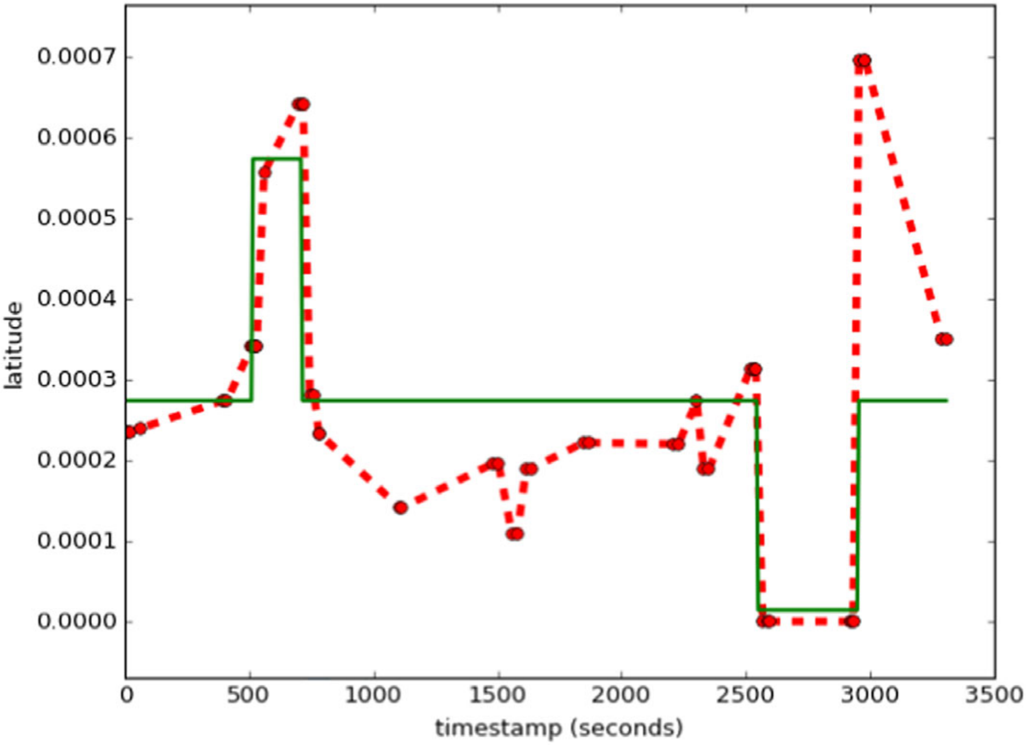

FIGURE 4 Reduced data set and approximation (Harvester 0)

where

$$\alpha_1 \overset{\text{def}}{=} \frac{h_1}{\hbar} = 1.5 \qquad \text{and} \qquad \alpha_2 \overset{\text{def}}{=} \frac{h_2}{\hbar} = 13.$$

This captures two of the excursions of the first harvester into the field as a linear combination of two functions of the form (3) and a reference latitude. The coefficients of the $B_{\hbar,T_i,w_i}$'s, i.e., the α_i's, are of order 1.

Roughly, we want to write the harvester's latitude track in the form

$$t \mapsto \sum_{(\hbar,T,w)\in\mathcal{U}'} \alpha_{(\hbar,T,w)} B_{\hbar,T,w}(t) + \hat{y}^* \tag{7}$$

for some finite subset $\mathcal{U}'$ of $\mathcal{U}$ and some coefficients $\{\alpha_{(\hbar,T,w)} : (\hbar,T,w) \in \mathcal{U}'\} \subset \mathbb{R}$. Although $\hat{y}^*$ of (1) was originally chosen as the edge-of-field latitude in (6), we would in fact like to algorithmically estimate y^* in (7).[3]

To proceed, we write things as an optimization problem in a vector space. Define the N-dimensional latitude observation vector

$$\mathbf{y} \overset{\text{def}}{=} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}.$$

Define also the constant N-dimensional column vector

$$\mathbf{1} \overset{\text{def}}{=} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}.$$

If a harvester stays at the edge of the field at some latitude y^*, his or her latitude at the observation times will be

$$\mathbf{1}y^* = \begin{pmatrix} y^* \\ y^* \\ \vdots \\ y^* \end{pmatrix}$$

at the t_n's. For each $(\hbar, T, w) \in \mathcal{U}$, define the $\mathbb{R}^N$-valued feature vector

$$\mathbf{f}_{\hbar,T,w} \overset{\text{def}}{=} \begin{pmatrix} B_{\hbar,T,w}(t_1) \\ B_{\hbar,T,w}(t_2) \\ \vdots \\ B_{\hbar,T,w}(t_N) \end{pmatrix}. \tag{8}$$

Using the latitudes at the t_n's as reference, (7) can be rewritten as

$$\mathbf{y} \approx \sum_{(\hbar,T,w)\in\mathcal{U}'} \alpha_{(\hbar,T,w)}\mathbf{f}_{\hbar,T,w} + \mathbf{1}y^*. \tag{9}$$

We want to find the “best” approximation of this type.

To proceed, we want to construct a *finite* subset $\mathcal{U}'$ of feature parameters. The height h of an excursion has already been restricted to $\hbar$. Since we can't really get much more granularity in the start of an excursion than the points in $\mathcal{T}$, we restrict T to the set $\mathcal{T}$. Although we could similarly restrict w so that the end of the excursion is also one of the elements of $\mathcal{T}$, this would make the $\mathcal{U}$ to be at least of size $\binom{N}{2}$. In fact, we will require that w take values in a reasonably small range; for specificity, we consider values of w in the set

$$\mathcal{W} \overset{\text{def}}{=} \{200, 300, 400, 500\}. \tag{10}$$

Namely, define

$$\mathcal{U}' \overset{\text{def}}{=} \{(\hbar, T, w) : w \in \mathcal{W}, T \in \mathcal{T}\}. \tag{11}$$

Then, $L \overset{\text{def}}{=} |\mathcal{U}'| = 4N$. Let

$$\left(\left(\hbar, T^o_\ell, w^o_\ell\right)\right)_{\ell=1}^{L} \tag{12}$$

be an ordering of $\mathcal{U}'$.

Our choice of $\mathcal{W}$ in (10) and subsequent choice of the finite set $\mathcal{U}'$ of (11) is empirically motivated by the data from Harvester 0 and the observation that the other harvesters, as seen from Figure 2 have behavior on similar scales. One could add more elements to $\mathcal{W}$, but at the risk of allowing more interference between excursions; see Section 5. Our choice of $\hbar$ of (5) is somewhat less important; variation in the reference height $\hbar$ could be subsumed into the choice of $\boldsymbol{\alpha}$.

We can now start to formulate an optimization problem corresponding to (9). We want to find the $(\alpha_\ell)_{\ell=1}^L \subset \mathbb{R}^L$ and $y^* \in \mathbb{R}$ which give the "best" approximation of the form

$$\mathbf{y} \approx \sum_{\ell=1}^{L} \alpha_\ell \mathbf{f}_{\hbar,t_\ell,w_\ell} + \mathbf{1}y^* = \mathbf{F}\boldsymbol{\alpha} + y^*\mathbf{1} = \mathbf{F}_e\begin{pmatrix}\boldsymbol{\alpha}\\ y^*\end{pmatrix}, \tag{13}$$

where

$$\mathbf{F} \overset{\text{def}}{=} \left(\mathbf{f}_{\hbar,t_1,w_1}\ \mathbf{f}_{\hbar,t_2,w_2}\ \cdots\ \mathbf{f}_{\hbar,t_L,w_L}\right),$$
$$\boldsymbol{\alpha} = \begin{pmatrix}\alpha_1\\ \alpha_2\\ \vdots\\ \alpha_L\end{pmatrix}, \tag{14}$$

and where $\mathbf{F}_e$ is the block matrix

$$\mathbf{F}_e \overset{\text{def}}{=} \left(\mathbf{F}\ \mathbf{1}\right).$$

To minimize the error of the approximation (13), we need to specify a way to measure the error. Some standard notation for norms on vector spaces will be useful here and for some other reasons. For any nonnegative integer M and any symmetric nonnegative $M \times M$ matrix Σ, define the seminorm

$$\|x\|_\Sigma^2 \overset{\text{def}}{=} x^T\Sigma x \tag{15}$$

for all $x \in \mathbb{R}^M$. The size of the error in (13) can now be measured via an $L^2(dt)$ norm. For $n \in \{2, 3 \dots N-1\}$, the difference between adjacent midpoints is

$$dt_n \overset{\text{def}}{=} \tfrac{1}{2}(t_{n+1}+t_n) - \tfrac{1}{2}(t_n + t_{n-1}) = \tfrac{1}{2}(t_{n+1} - t_{n-1});$$

define

$$D = \operatorname{diag}\left(t_2 - t_1\ dt_2\ dt_3\ \cdots\ dt_{N-1}\ t_N - t_{N-1}\right).$$

The error of (13) can now be measured as

$$\left\|\mathbf{y} - \left\{\mathbf{F}\boldsymbol{\alpha} + \mathbf{1}y^*\right\}\right\|_D^2 = \left\|\mathbf{y} - \mathbf{F}_e\begin{pmatrix}\boldsymbol{\alpha}\\ y^*\end{pmatrix}\right\|_D^2. \tag{16}$$

The $L^2(dt)$ norm clumps together readings at very similar times (as in the clump of data shortly after time 16,000 in Figure 3).

We can of course directly minimize (16) over all $\boldsymbol{\alpha}$ and y^*. Define

$$\begin{pmatrix}\boldsymbol{\alpha}_0\\ y_0^*\end{pmatrix} = \left(\mathbf{F}_e^T\mathbf{F}_e\right)^{-1}\mathbf{F}_e^T\mathbf{y}. \tag{17}$$

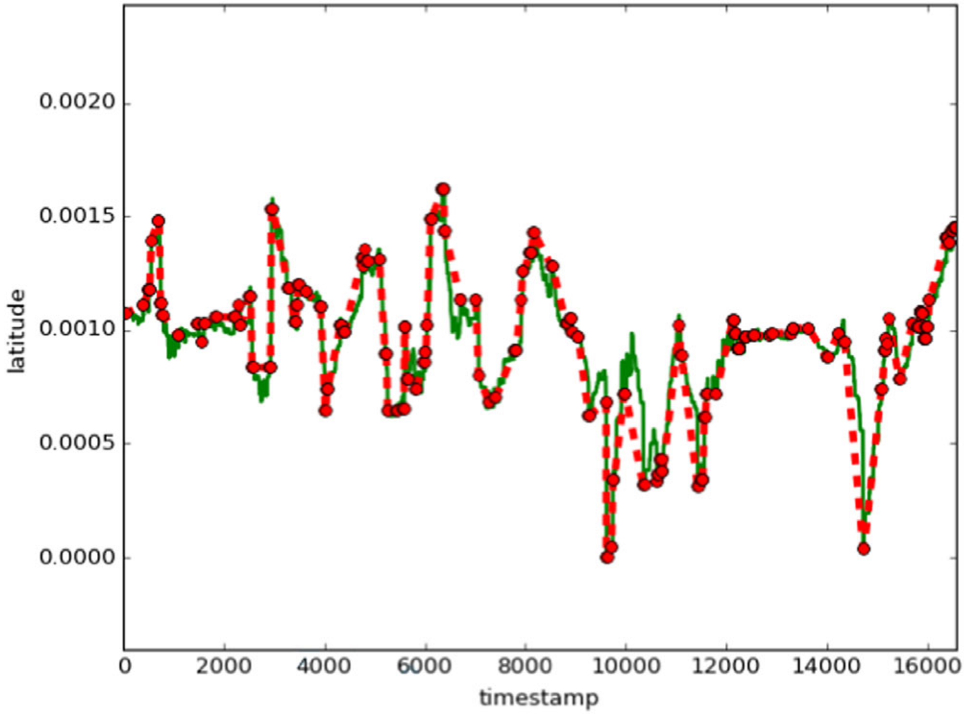

FIGURE 5 $L^2(dt)$ approximation

This leads to a fairly good approximation of the latitude track; see Figure 5. This is not surprising given the dimensionality of the feature set of $\mathcal{U}'$. However, the α_e's do not allow us to easily distinguish large and macroscopically visible features from noise. The downward excursion between times 14,000 and 16,000 in Figure 5 is approximated by a series of smaller excursions (note the green steps). Indeed, Figure 6 suggests something of a normal distribution of excursion height. In fact, we want the feature size to be more multimodal with clustering at sizes of order $\pm\hbar$.

In fact, we want $\boldsymbol{\alpha}$ to have only a few nonzero components; we want to enforce *sparsity*. Ideally, we would like to penalize the L^0 norm on the components of $\boldsymbol{\alpha}$; recall that

$$\|\boldsymbol{\alpha}\|_0 \overset{\text{def}}{=} \left|\left\{\ell : \alpha_\ell \neq 0\right\}\right|$$

for $\boldsymbol{\alpha} = (\alpha_1, \alpha_2 \dots \alpha_L)$, where $|A|$ is the cardinality of A. Fix a penalty parameter $\varkappa > 0$ and consider the minimization problem

$$\min_{y^* \in \mathbb{R}, \boldsymbol{\alpha} \in \mathbb{R}^L} \mathcal{E}(y^*, \boldsymbol{\alpha}), \tag{18}$$

where

$$\mathcal{E}(y^*, \boldsymbol{\alpha}) \overset{\text{def}}{=} \left\|\mathbf{y} - \mathbf{F}\boldsymbol{\alpha} - \mathbf{1}y^*\right\|_D^2 + \frac{1}{2}\varkappa \|\boldsymbol{\alpha}\|_0 . \tag{19}$$

This is now a well-understood problem; there are a number of ways to regularize this problem and enforce sparsity; the standard approach depends on least absolute shrinkage and selection operator (LASSO) techniques; see Hastie, Tibshirani, and Wainwright (2015). We would like

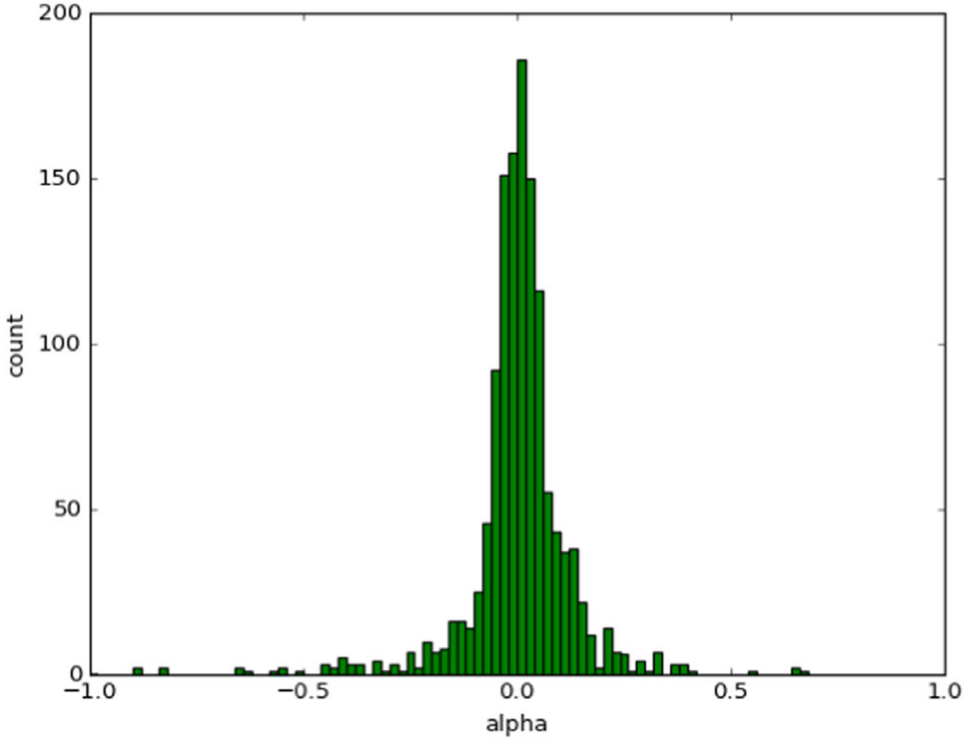

FIGURE 6 Histogram of components of $\boldsymbol{\alpha}_0$

to, however, try to adapt the recent work of Liu and Li (2016), which suggests something like an expectation-maximization (EM) algorithm to regularize (18). Note, however, that we want to be careful to *not* penalize a nonzero y^*; the calculations of Liu and Li (2016) must be slightly modified.

To incorporate the thoughts of Liu and Li (2016), we approximate $\mathcal{E}$ to separate out some effects of feature inclusion or exclusion. For $\mathbf{m} = (m_1, m_2 \ldots m_L) \in \mathbb{R}^L$, define

$$\begin{aligned}
\Pi_{\mathbf{m}} &\overset{\text{def}}{=} \operatorname{diag}\left(\chi_{\{m_1 \neq 0\}} \; \chi_{\{m_2 \neq 0\}} \; \cdots \; \chi_{\{m_L \neq 0\}} \right) \\
S_{\mathbf{m}}^{-} &\overset{\text{def}}{=} \operatorname{diag}\left(m_1^{-2} \chi_{\{m_1 \neq 0\}} \; m_2^{-2} \chi_{\{m_2 \neq 0\}} \; \cdots \; m_L^{-2} \chi_{\{m_L \neq 0\}} \right) \\
S_{\mathbf{m}}^{+} &\overset{\text{def}}{=} \operatorname{diag}\left(m_1^2 \; m_2^2 \; \cdots \; m_L^2 \right);
\end{aligned}$$

note that all of these matrices are symmetric and nonnegative semidefinite. Also, note that $\Pi_{\mathbf{m}}$ is a projection operator and that specifically

$$\left(\Pi_{\mathbf{m}} \boldsymbol{\alpha}\right)_\ell = \begin{cases} \alpha_\ell & \text{if } m_\ell \neq 0 \\ 0 & \text{if } m_\ell = 0. \end{cases}$$

Define

$$\begin{aligned}\hat{\mathcal{E}}(y^*,\boldsymbol{\alpha},\mathbf{m}) &\overset{\text{def}}{=} \left\|\mathbf{y}-\mathbf{F}\Pi_{\mathbf{m}}\boldsymbol{\alpha}-\mathbf{1}y^*\right\|_D^2+\frac{\varkappa}{2}\sum_{\ell:\alpha_\ell\neq 0}\frac{\alpha_\ell^2}{m_\ell^2}\\ &= \left\|\mathbf{y}-\mathbf{F}\Pi_{\mathbf{m}}\boldsymbol{\alpha}-\mathbf{1}y^*\right\|_D^2+\frac{\varkappa}{2}\left\|\boldsymbol{\alpha}\right\|_{S_{\mathbf{m}}^-}^2\\ &= \left\|\mathbf{y}-\mathbf{F}\Pi_{\mathbf{m}}\boldsymbol{\alpha}-\mathbf{1}y^*\right\|_D^2+\frac{\varkappa}{2}\left\|\Pi_{\mathbf{m}}\boldsymbol{\alpha}\right\|_{S_{\mathbf{m}}^-}^2\end{aligned} \tag{20}$$

for $y^*\in\mathbb{R}$ and $\boldsymbol{\alpha}$ and $\mathbf{m}$ in $\mathbb{R}^L$. The last line uses the fact that $\Pi_{\mathbf{m}}^T S_{\mathbf{m}}^-\Pi_{\mathbf{m}}=S_{\mathbf{m}}^-$. Note that

$$\mathcal{E}(y^*,\boldsymbol{\alpha})=\hat{\mathcal{E}}(y^*,\boldsymbol{\alpha},\boldsymbol{\alpha}).$$

We then want to minimize $\hat{\mathcal{E}}$ over y^* and $\boldsymbol{\alpha}$ with the added constraint that $\mathbf{m}=\boldsymbol{\alpha}$. We want to do this via a recursive algorithm. Fix an initial $y_0^*\in\mathbb{R}$ and $\boldsymbol{\alpha}_0\in\mathbb{R}^L$ and iteratively define

$$(y_{n+1}^*,\boldsymbol{\alpha}_{n+1})\overset{\text{def}}{=}\operatorname*{arg\,min}_{\substack{\hat{y}^*\in\mathbb{R}\\ \hat{\boldsymbol{\alpha}}\in\mathbb{R}^L}}\hat{\mathcal{E}}(\hat{y}^*,\hat{\boldsymbol{\alpha}},\boldsymbol{\alpha}_n) \tag{21}$$

for $n\in\{0,1\,\dots\}$.

5 | THE ALGORITHM

The solution of (21) is fairly simple; the extremals are easy to compute. Note, by way of prelude, that the gradient of (15) at a point $x\in\mathbb{R}^M$ is given by $2\Sigma x$. For a fixed $\mathbf{m}\in\mathbb{R}^L$, differentiating the map $(\hat{y}^*,\hat{\boldsymbol{\alpha}})\mapsto\hat{\mathcal{E}}(\hat{y}^*,\hat{\boldsymbol{\alpha}},\mathbf{m})$ with respect to $\hat{\boldsymbol{\alpha}}$ and $\hat{y}^*$, we thus have that

$$\begin{aligned}-\left(\mathbf{y}-\mathbf{F}\Pi_{\mathbf{m}}\boldsymbol{\alpha}-\mathbf{1}y^*\right)^T D\mathbf{F}\Pi_{\mathbf{m}}+\varkappa\boldsymbol{\alpha}^T S_{\mathbf{m}}^-&=0\\ -\left(\mathbf{y}-\mathbf{F}\Pi_{\mathbf{m}}\boldsymbol{\alpha}-\mathbf{1}y^*\right)^T D\mathbf{1}&=0,\end{aligned}$$

respectively. Rearranging,

$$\begin{aligned}\left(\Pi_{\mathbf{m}}\mathbf{F}^T D\mathbf{F}\Pi_{\mathbf{m}}+\varkappa S_{\mathbf{m}}^-\right)\boldsymbol{\alpha}+\Pi_{\mathbf{m}}\mathbf{F}^T D\mathbf{1}y^* &= \Pi_{\mathbf{m}}\mathbf{F}^T D\mathbf{y}\\ \mathbf{1}^T D\mathbf{F}\Pi_{\mathbf{m}}\boldsymbol{\alpha}+\mathbf{1}^T D\mathbf{1}y^* &= \mathbf{1}^T D\mathbf{y}.\end{aligned} \tag{22}$$

Multiply the first equation by $S_{\mathbf{m}}^+$ and note that $S_{\mathbf{m}}^+\Pi_{\mathbf{m}}=S_{\mathbf{m}}^+$ and $S_{\mathbf{m}}^+S_{\mathbf{m}}^-=\Pi_{\mathbf{m}}$. Letting $\mathbf{I}_L$ be the $L\times L$ identity matrix,

$$\left(S_{\mathbf{m}}^+\mathbf{F}^T D\mathbf{F}+\varkappa\mathbf{I}_L\right)\Pi_{\mathbf{m}}\boldsymbol{\alpha}+S_{\mathbf{m}}^+\mathbf{F}^T D\mathbf{1}y^*=S_{\mathbf{m}}^+\mathbf{F}^T D\mathbf{y}. \tag{23}$$

Note that this depends on $\boldsymbol{\alpha}$ through $\Pi_{\mathbf{m}}\boldsymbol{\alpha}$. The last equation of (20) implies that $\hat{\mathcal{E}}(y^*,\boldsymbol{\alpha},\mathbf{m})$ does not depend on $(\mathbf{I}_L-\Pi_{\mathbf{m}})\boldsymbol{\alpha}$, so let's add the requirement that

$$\left(\mathbf{I}_L-\Pi_{\mathbf{m}}\right)\boldsymbol{\alpha}=0.$$

Adding this to the second equation of (22) and to (23),

$$\begin{aligned}\left(S_{\mathbf{m}}^{+}\mathbf{F}^{T}D\mathbf{F}+\varkappa\mathbf{I}_{L}\right)\boldsymbol{\alpha}+S_{\mathbf{m}}^{+}\mathbf{F}^{T}D\mathbf{1}y^{*}&=S_{\mathbf{m}}^{+}\mathbf{F}^{T}D\mathbf{y}\\ \mathbf{1}^{T}D\mathbf{F}\boldsymbol{\alpha}+\mathbf{1}^{T}D\mathbf{1}y^{*}&=\mathbf{1}^{T}D\mathbf{y}.\end{aligned}$$

In block form,

$$\begin{pmatrix}\left(S_{\mathbf{m}}^{+}\mathbf{F}^{T}D\mathbf{F}+\varkappa\mathbf{I}_{L}\right) & S_{\mathbf{m}}^{+}\mathbf{F}^{T}D\mathbf{1}\\ \mathbf{1}^{T}D\mathbf{F} & \mathbf{1}^{T}D\mathbf{1}\end{pmatrix}\begin{pmatrix}\boldsymbol{\alpha}\\ y^{*}\end{pmatrix}=\begin{pmatrix}S_{\mathbf{m}}^{+}\mathbf{F}^{T}D\mathbf{y}\\ \mathbf{1}^{T}D\mathbf{y}\end{pmatrix}. \tag{24}$$

In fact, we can further rewrite this. Define the block matrices

$$\mathbf{S}_{\mathbf{m}}\overset{\text{def}}{=}\begin{pmatrix}S_{\mathbf{m}}^{+} & 0\\ 0 & 1\end{pmatrix}$$

$$\hat{\mathbf{I}}\overset{\text{def}}{=}\begin{pmatrix}\mathbf{I}_{L} & 0\\ 0 & 0\end{pmatrix}$$

$$\mathbf{A}\overset{\text{def}}{=}\mathbf{F}_{e}^{T}D\mathbf{F}_{e}$$

$$\mathbf{b}\overset{\text{def}}{=}\mathbf{F}_{e}^{T}D\mathbf{y}.$$

Then, (24) is

$$\left(\mathbf{S}_{\mathbf{m}}\mathbf{A}+\varkappa\hat{\mathbf{I}}\right)\begin{pmatrix}\boldsymbol{\alpha}\\ y^{*}\end{pmatrix}=\mathbf{S}_{\mathbf{m}}\mathbf{b}.$$

Thus,

$$\begin{pmatrix}\boldsymbol{\alpha}\\ y^{*}\end{pmatrix}=\left(\mathbf{S}_{\mathbf{m}}\mathbf{A}+\varkappa\hat{\mathbf{I}}\right)^{-1}\mathbf{S}_{\mathbf{m}}\mathbf{b}.$$

The recursive algorithm is then

$$\begin{pmatrix}\boldsymbol{\alpha}_{n+1}\\ y_{n+1}^{*}\end{pmatrix}=\left(\mathbf{S}_{\alpha_{n}}\mathbf{A}+\varkappa\hat{\mathbf{I}}\right)^{-1}\mathbf{S}_{\alpha_{n}}\mathbf{b}.$$

This recursion is started at (17).

5.1 | Termination threshold

Fix now a termination threshold

$$\boxed{\boxed{\tau\overset{\text{def}}{=}0.01}} \tag{25}$$

and stop the recursion at the first N^{*} such that

$$\|\boldsymbol{\alpha}_{N^{*}+1}-\boldsymbol{\alpha}_{N^{*}}\|_{L^{2}}\leq\tau\|\boldsymbol{\alpha}_{N^{*}}\| \tag{26}$$

(i.e., when the change in $\boldsymbol{\alpha}_{n}$ is less than 1% of $\boldsymbol{\alpha}_{n}$ itself). From some numerical exploration, $N^{*}<30$ in most cases.

5.2 | Quantization threshold

In a sense, the relaxed L^0 algorithm of Section 5 is designed to minimize the number of "nonsmall" components of $\boldsymbol{\alpha}$. After the algorithm has terminated, we forcibly set the "small" elements of $\boldsymbol{\alpha}_{N^*}$ to zero. The term $(\alpha_{N^*})_\ell \mathbf{f}_{\hbar,t_\ell,w_\ell}$ corresponds to an excursion of height $(\boldsymbol{\alpha}_{N^*})_\ell \hbar$. Let's assume that the algorithm of Section 5 indeed thinks that an excursion is in fact spurious if it is less than 0.00002° latitude; i.e., if $|(\boldsymbol{\alpha}_{N^*})_\ell| < \delta$, where

$$\boxed{\boxed{\delta \overset{\text{def}}{=} 0.01.}}. \tag{27}$$

Define $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^L$ as

$$\hat{\boldsymbol{\alpha}}_\ell \overset{\text{def}}{=} \begin{cases} (\boldsymbol{\alpha}_{N^*})_\ell & \text{if } \left|(\boldsymbol{\alpha}_{N^*})_\ell\right| \geq \delta \\ 0 & \text{else.} \end{cases} \tag{28}$$

6 | PENALTY SIZE

In order for (18) (or more accurately (21)) to decompose the data $\mathbf{y}$ into a small number of $\mathbf{f}_{\hbar,T,w}$ pattern vectors as in (8), we want to choose $\varkappa$ so that the penalty term in (19) is sufficiently strong compared to the mean-square error of the approximation (the $\|\cdot\|_D^2$ term in (18) or (21)). Latitude measurements have error of size 10^{-8} while the time instants are in seconds. Over a time interval of size T (in seconds)

$$\|\mathbf{y} - \mathbf{F}\boldsymbol{\alpha} - \mathbf{1}y^*\|_D^2 = O(10^{-16}T).$$

The excursions are on the order of 5 minutes or 300 seconds, so in time T we should have $T/300$ excursions; thus,

$$\tfrac{1}{2}\varkappa \, \|\boldsymbol{\alpha}\|_0 = \varkappa O(10^{-2}T);$$

matching these up, we start by requiring that $\varkappa \gg O(10^{-14})$.

Figures 7 and 8 show the approximation for small (10^{-10}) and large (10^{-5}) values of $\varkappa$; in these figures, we have not thresholded as in (28) (or alternately, we have set $\delta = 0$). Before thresholding as in (28), we would like to see that the algorithm of (5) identifies a few large excursions. For $\varkappa = 10^{-5}$, the iteration given by (25) and (26) stops after 7 iterations. If $\varkappa = 10^{-10}$, the rule of (25) and (26) does not stop the iteration; α seems not to converge. Figure 9 shows that $\|d\alpha\|/\|\alpha\|$ seems to oscillate around 0.5. On the other hand, if $\varkappa = 10^{-5}$, almost all of the $(\alpha_{N^*})_\ell$'s are small (we decompose a large excursion into a large number of really small excursions); we would like to decompose the latitude track in to a few macroscopically visible features.

Some numerical exploration reveals that

$$\boxed{\boxed{\varkappa = 1.5 \times 10^{-7}}} \tag{29}$$

gives reasonably good performance, and stops after 8 iterations; see Figure 10.

In Figures 7, 8, and 10, the blue dots represent the sizes and starting points of different features. Adding these features up gives the green approximate excursion paths.

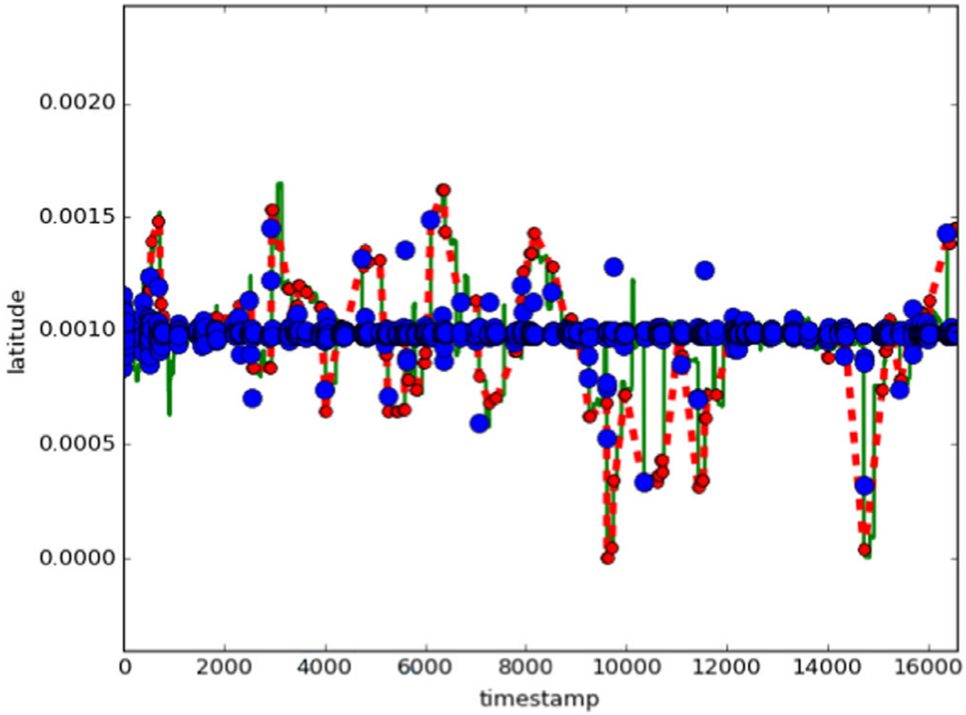

FIGURE 7 Approximation for Harvester 0 with $\varkappa = 10^{-10}$ after 30 iterations

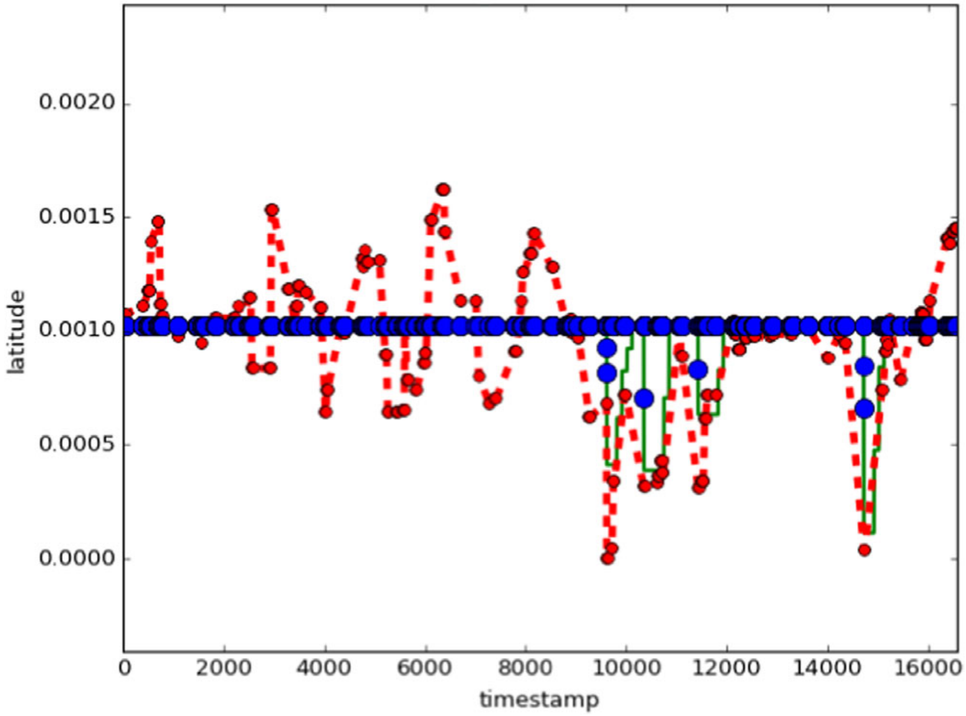

FIGURE 8 Approximation for Harvester 0 with $\varkappa = 10^{-5}$

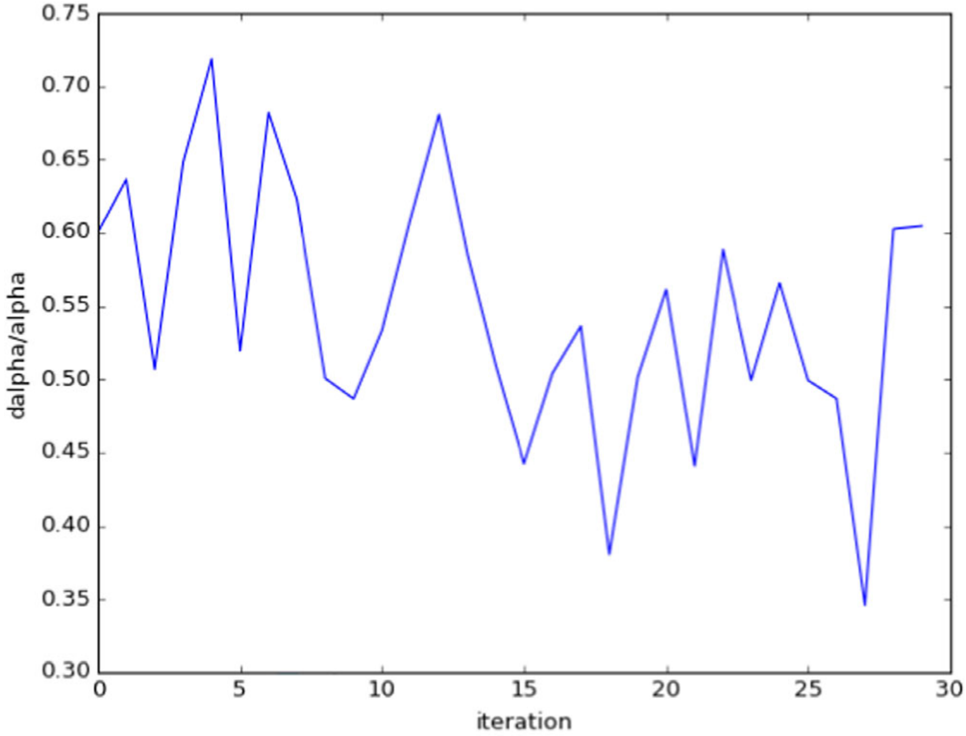

FIGURE 9 $\|d\alpha\|/\|\alpha\|$ for $\varkappa = 10^{-10}$

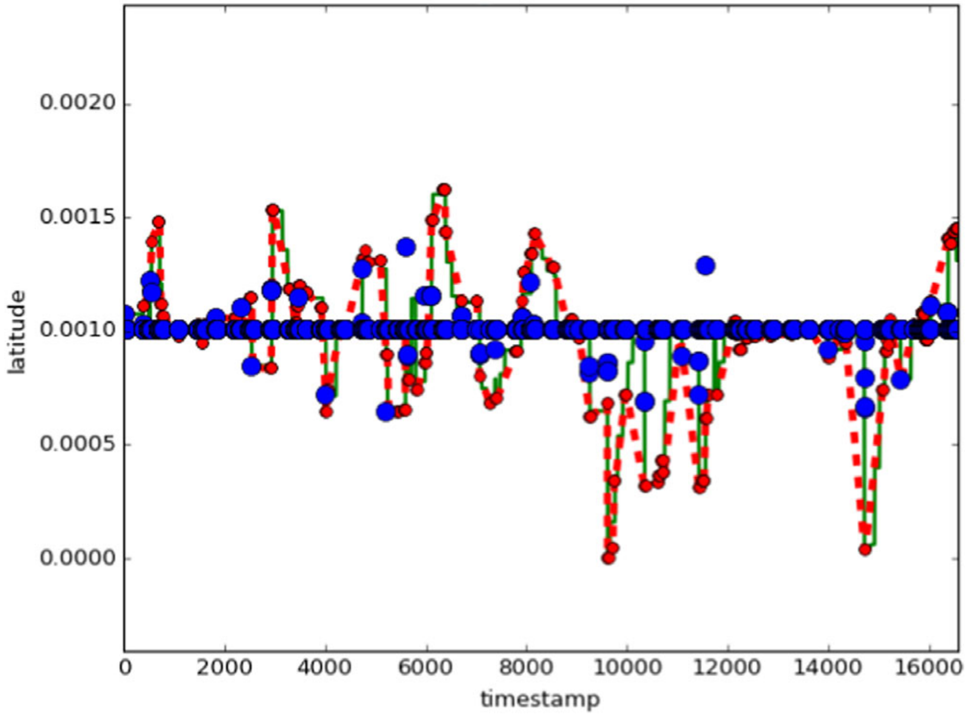

FIGURE 10 Approximation for Harvester 0 with $\varkappa = 1.5 \times 10^{-7}$

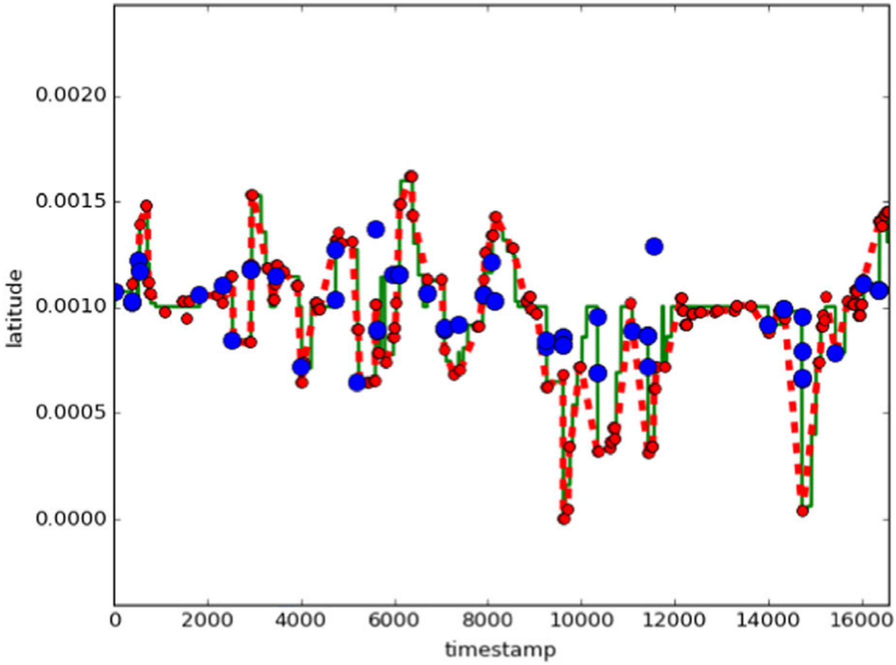

FIGURE 11 Approximation for Harvester 0 with $\varkappa = 1.5 \times 10^{-7}$; features thresholded at $\delta = 0.01$

We now threshold the data from Figure 10, as in (27) and (28). The result is Figure 11.

7 | OVERLAPPING EXCURSIONS

Figure 10 shows another problem; the above algorithm tends to decompose an excursion with several data points at different latitudes into multiple overlapping excursions (i.e., multilevel excursions). Near timestamp 10, 000 in Figure 10, some multilevel excursions have been decomposed into several overlapping features. This causes a challenge for the L^0 algorithm we are using. While we might be tempted to increase $\varkappa$ to further penalize these overlapping excursions, the size (about 0.0003° degrees of latitude) of these multilevel excursions may in fact be the same as some of the smaller excursions (see those near timestamp 4000 in Figure 10). Increasing $\varkappa$ could just as well remove the approximation of the excursion near timestamp 4000 as remove the second level of an excursion near timestamp 10, 000.

We resolve this problem by *combining* excursions; we will do this *after* the iterations of (21) and Section 5 iterate (i.e., after the thresholds of subsection 5.1 have been reached). We create a partial order on $\mathcal{U}$ by stating that the excursion $(h_1, T_1, w_1) \leq (h_2, T_2, w_2)$ if

$$T_2 \leq T_1 \leq T_2 + w_2,$$

then B_{h_1,T_1,w_1} starts during B_{h_2,T_2,w_2}. This leads to a multilevel excursion. In this case, we add the height of the excursion with parameter (h_1, T_1, w_1) to the excursion (h_2, T_2, w_2) and remove the excursion at (h_1, T_1, w_1). The resulting excursion still has support $(T_2, T_2 + w_2)$ but has height, which reflects the combination of the excursions. While some data are thus lost, the beginning of the excursion, i.e., T_2 (which, hopefully, respects when and where the harvester entered the field), is preserved.

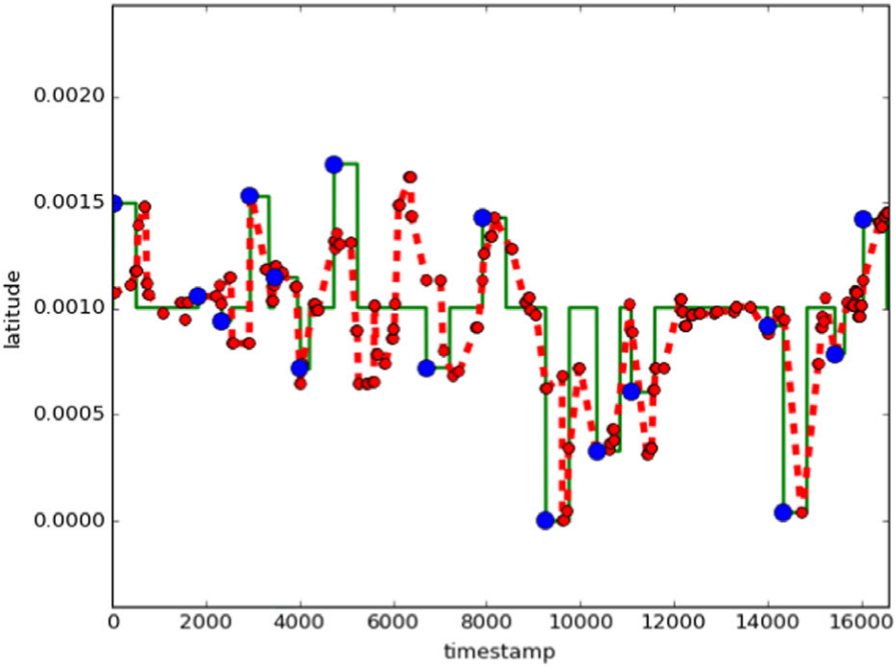

FIGURE 12 Approximation for Harvester 0 with $\varkappa = 1.5 \times 10^{-7}$; features thresholded at $\delta = 0.01$ and combined

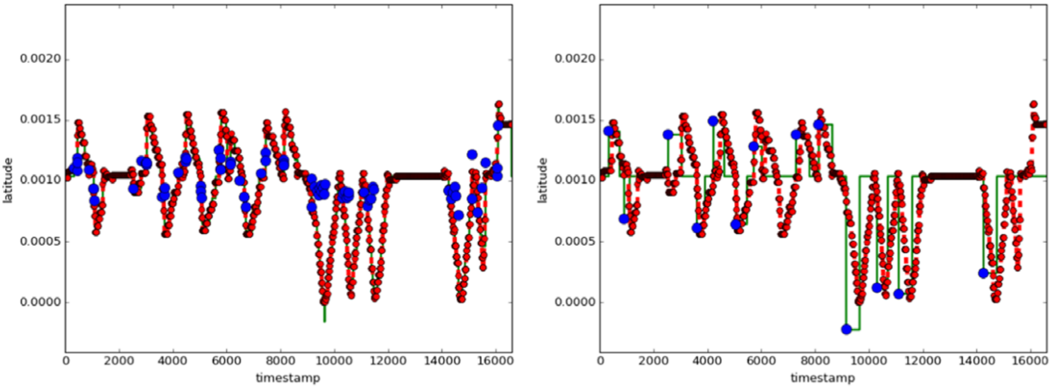

FIGURE 13 Approximation for Harvester 1 with $\varkappa = 1.5 \times 10^{-7}$; features thresholded at $\delta = 0.01$; excursions uncombined (left) and combined (right)

This step requires a large number of pairwise comparisons of nonzero excursions. Naïvely, one might look at all pairs of nonzero excursions, combine them if they are ordered, and then repeat until no more combinations can be done. In fact, this can be simplified if the universe (2) of possible excursions is organized when constructing **F** of (14). Choose $\mathcal{U}'$ of (11) as follows:

$$\begin{gathered}\{(\hbar, T_1, 500), (\hbar, T_2, 500) \ldots (\hbar, T_N, 500), \cdots \\ (\hbar, T_1, 400), (\hbar, T_2, 400) \ldots (\hbar, T_N, 400) \cdots \\ (\hbar, T_1, 200), (\hbar, T_2, 200) \ldots (\hbar, T_N, 200)\}\end{gathered}$$

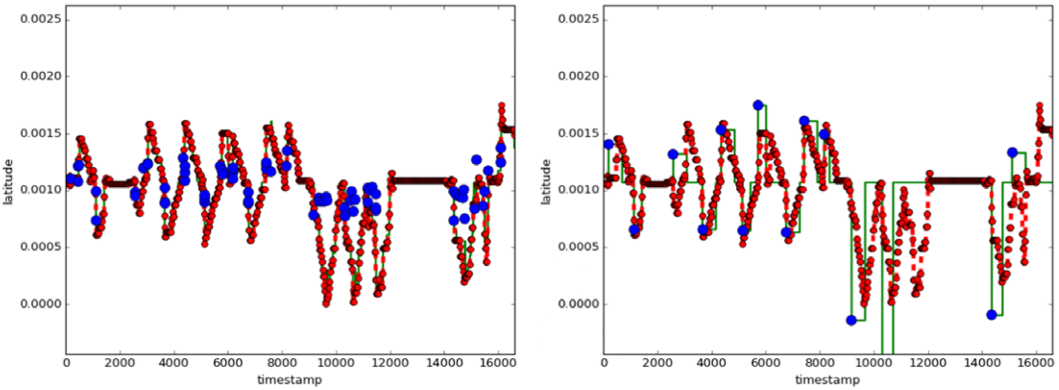

FIGURE 14 Approximation for Harvester 2 with $\varkappa = 1.5 \times 10^{-7}$; features thresholded at $\delta = 0.01$; excursions uncombined (left) and combined (right)

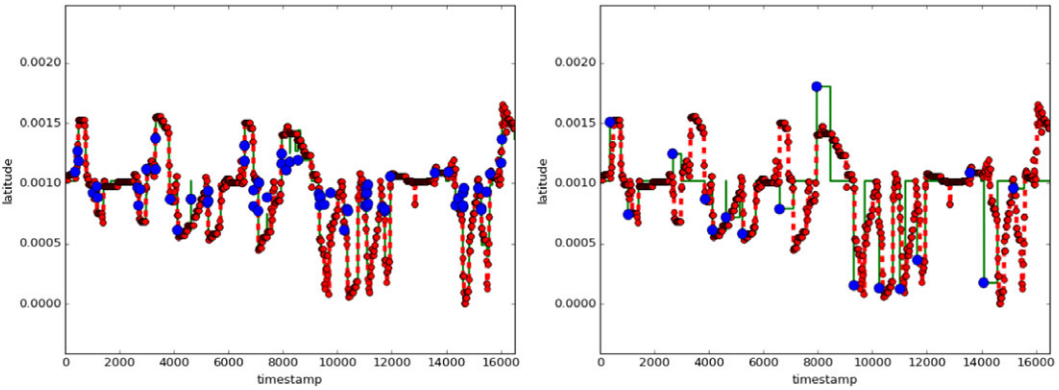

FIGURE 15 Approximation for Harvester 3 with $\varkappa = 1.5 \times 10^{-7}$; features thresholded at $\delta = 0.01$; excursions uncombined (left) and combined (right)

(i.e., in the notation of (12), $(\hbar, T_1^o, w_1^o) = (\hbar, T_1, 500)$, $(\hbar, T_2^o, w_2^o) = (\hbar, T_2, 500)$, $(\hbar, T_{N+1}^o, w_{N+1}^o) = (\hbar, T_1, 400)$, and so forth). We can then sequence through $\mathcal{U}'$ and combine excursions. Sequence through increasing ℓ, and sequence through decreasing ℓ' with $\ell \le \ell'$, and combine $(\hbar, T_\ell^o, w_\ell^o)$ and $(\hbar, T_{\ell'}^o, w_{\ell'}^o)$ if $(\hbar, T_\ell^o, w_\ell^o) \le (\hbar, T_{\ell'}^o, w_{\ell'}^o)$. This combines overlapping excursions, with smaller excursions being folded into larger ones.

There are now 16 excursions; see Figure 12.

8 | RESULTS AND DISCUSSION

To understand the performance of the above algorithm, we apply our algorithm to the other harvesters;

- set $\varkappa = 1.5 \times 10^{-7}$ (as in (29)),
- stop the iteration when $\|d\alpha\|_1 \le 0.1\|\alpha\|_1$ (as in (25) and (26)),
- threshold the α_ℓ's at 0.01 as in (27) and (28).

There are several ways in which the algorithm can fail.

The plot for Harvester 1 is in Figure 13. The algorithm misses several excursions near time $t = 5000$. Based on the uncombined excursions, the combination procedure combines together some adjacent positive and negative excursions. Similar behaviors hold for Harvesters 2 and 3; see Figures 14 and 15. More granularity in the collection of widths of (10) would allow better representation of the transition between negative and positive excursions.

A related problem is apparent in Figure 15 (and in particular, the right-hand plot in Figure 15, where the features have been combined). The negative excursion that starts at around time 4000 is broken into two smaller negative excursions. Here, adding longer excursions to (10) would algorithmically allow for this.

9 | CONCLUSION

The calculations of this paper, while still allowing room for improvement, do allow one to identify *when* and *where* harvesting is taking place. This could lead to more precise yield maps, which give information about field productivity on a reasonably granular scale. In particular, precise information about harvest, such as we have attempted, could lead to row-by-row productivity.

In practice, harvesters make excursions into a field to harvest a prespecified amount of fruit; in the case of strawberries, a “crate” of several “clamshell” containers (these clamshells are what ultimately appear in retail stores). Once these crates are filled, they are subjected to a quality check (at the edge of the field). Timestamps from these quality checks translate into when a crate of berries has been harvested; combining that with the ability to identify excursions into fields, as we have proposed here, allows one to identify where and at what rate a crate has been harvested. This leads to yield maps.

A further exploration of the ideas proposed in this work might depend on a larger data set with a wider range of possible behaviors. It would also be important to compare the results of our algorithm with a ground truth of actual harvesting records. In order to implement some of these harvest estimates at a larger level, some understanding of reliability would be needed.

ACKNOWLEDGMENTS

Thanks to Maureen McGuire for allowing us to work with Crisalida Farms. This work was partly supported by the UIUC Campus Research oard.

ENDNOTES

[1] The video https://www.youtube.com/watch?v=CFORJwDMxqs gives a realistic idea of harvesting.

[2] See http://www.ers.usda.gov/topics/farm-economy/farm-labor/background.aspx

[3] In fact, most farms do have precise GPS data about field boundaries, meaning that y^* is already known. However, harvesting crews can, during the course of a day, finish one field and move to another. There are thus a finite number of possible field edges, and there is a natural change point detection problem associated with algorithmically identifying when a crew moves to a new field. Algorithmically identifying the edge of the field gives us some preliminary tools to understanding variability in the field itself.

REFERENCES

Ellis, K., Godbole, S., Marshall, S., Lanckriet, G., Staudenmayer, J., & Kerr, J. (2014). Identifying active travel behaviors in challenging environments using GPS, accelerometers, and machine learning algorithms. *Public Health*, *2*, 39–46.

Gemtos, T., Fountas, S., Tagarakis, A., & Liakos, V. (2013). Precision agriculture application in fruit crops: Experience in handpicked fruits. *Procedia Technology*, *8*, 324–332.

Hastie, T., Tibshirani, R., & Wainwright, M. (2015). *Statistical learning with sparsity: The LASSO and generalizations*. Boca Raton: CRC press.

Liu, Z., & Li, G. (2016). Efficient regularized regression with penalty for variable selection and network construction, *Computational and Mathematical Methods in Medicine*. https://www.hindawi.com/journals/cmmm/2016/3456153/cta/

Manuel, D., & Sowers, R. B. (2017). Optimal transport to cold chain in perishable hand-picked agriculture. *Natural Resource Modeling*, *30*, 2. https://onlinelibrary.wiley.com/doi/10.1111/nrm.12124/full

Rossi, L., Walker, J., & Mirco, M. (2015). Spatio-temporal techniques for user identification by means of GPS mobility data. *EPJ Data Science*, *4*(1), 1. https://link.springer.com/article/10.1140/epjds/s13688-015-0049-x

United Nations. (2015). *World population prospects, 2015 revision*. United Nations, New York.

Zheng, Y., Li, Q., Chen, Y., Xie, X., & Ma, W.-Y. (2008). *Understanding mobility based on GPS data*, Proceedings of the 10th International Conference on Ubiquitous Computing, ACM, pp. 312–321.

How to cite this article: Srivastava N, Maneykowski P, Sowers RB. Algorithmic geolocation of harvest in hand-picked agriculture. *Natural Resource Modeling*. 2018;31:e12158. https://doi.org/10.1111/nrm.12158