MACHINE LEARNING AND TASK DISAMBIGUATION IN HAND-PICKED AGRICULTURE

BY

NITIN SRIVASTAVA

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Industrial Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Adviser:

Professor Richard Sowers

Abstract

Although GPS-based travel data has been studied by many mainly for automated travel mode detection, the area of activity mode detection during harvest still remains an open technical challenge. This thesis proposes and tests a pattern recognition approach to harvest mode recognition from GPS travel data collected from 4 volunteers for 2 days in Oxnard, California. Three profiles were created to characterize activities performed during harvest. Piecewise quadratic interpolation was used on smoothened data to detect segments in trips taken by workers. Trip segments are then evaluated with the different profiles to find the best fitting profiles and the associated optimal parameters. Results indicated that the proposed framework performs well under data discrepancies. Identification of different modes during harvest is of relevance for assessing productivity of different workers and addressing any mismatch in vehicle scheduling. In our assessment, this proof-of-principle study demonstrates a use case for using GPS data in disambiguating different activities conducted during harvest; scalability of the methodology remains a challenge - programming GPUs to take advantage of independence in the different processes has been proposed to reduce the code runtime.

Acknowledgements

I would like to extend my sincere gratitude to my advisor, Professor Richard Sowers and Crisalida Farms, Oxnard - California for giving me the opportunity to work on a topic which is in the field of my passion and for guiding me through the research with their expertise and valuable inputs during the course of the project as well as in writing this thesis. I would like to thank my family for their support while earning this Master's degree.

Table of Contents

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 of 7.3 billion [1]. Demand for staple food-grains like cereals is expected to reach 3 billion tons in 2050 from 2.1 billion tons in 2009 [2]. A combination of increase in the demand of food crops with a sub-par growth in arable land means that feeding a world population of 9.1 billion people in 2050 [2] would require raising overall food production by ~ 70% between 2005/07 and 2050 [2], (see also Figure 1). It has become clear that the efficiency in all types of food production is of increasing importance, both economically and in the interest of sustainability.

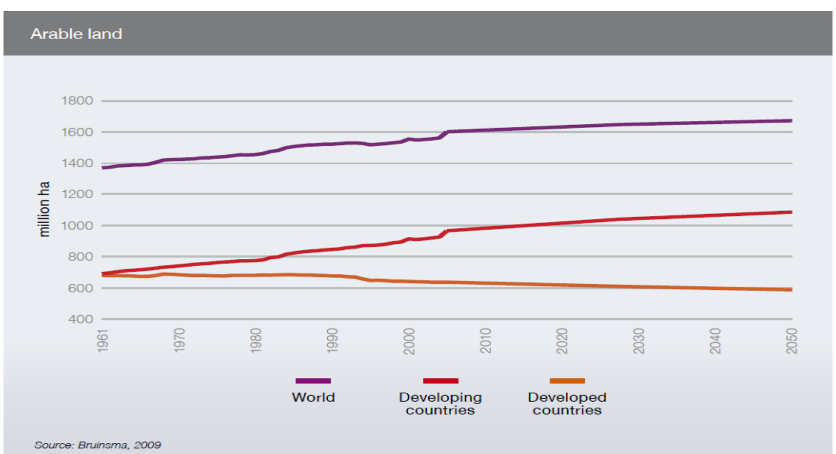


Figure 1 – Forecast of arable land in the world (In Mil. Hectare)

How do various environmental factors (temperature, irrigation, fertilizer, time of planting, etc.) affect growth? When is the best time to harvest? How can problems related to mineral variability in fields be quickly identified and resolved? If the promise of data analytics can be brought to bear on these challenges, the benefits, both in terms of economics and sustainability, would be immense.

Precision agriculture has been naturally driven by various technological advances in how we store, transmit and process data. Increasingly ubiquitous sensing abilities e.g., soil, drone, and satellite (see Figure 2) allow precise understanding of what goes in and how it affects the state of the field.

For row crops (wheat, corn, soy, etc.), large mechanized farming equipment provide 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 (see Figure 2).

The promise of all of this 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.

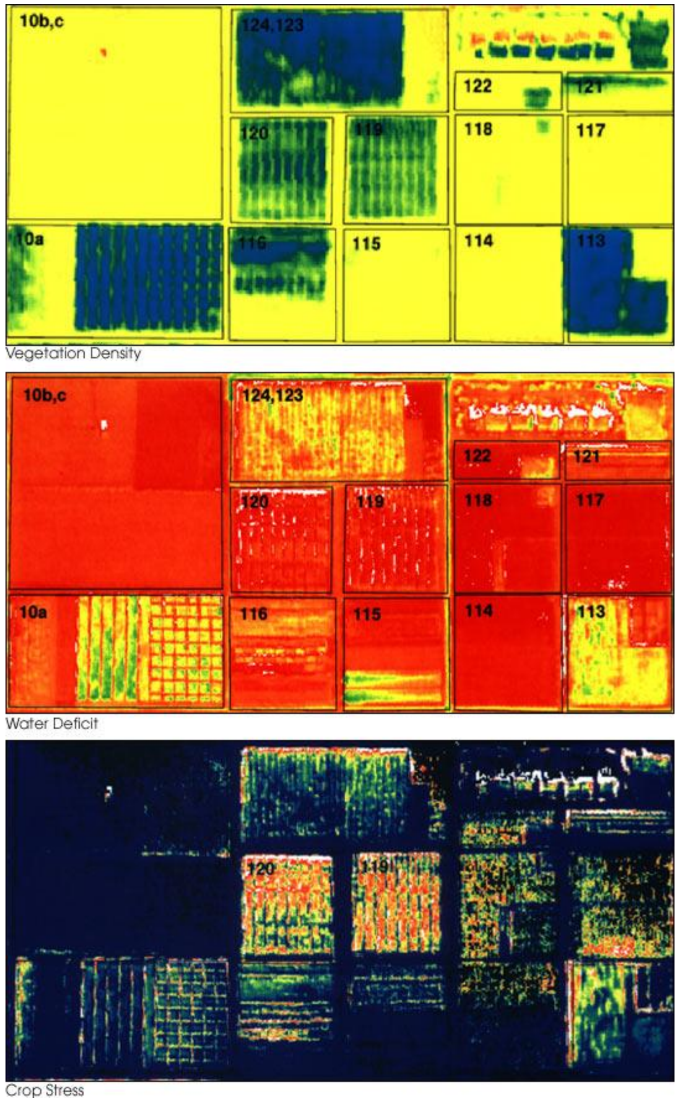


Figure 2 - The images were acquired by the Daedalus sensor aboard a NASA aircraft flying over the Maricopa Agricultural Center in Arizona.

- The top image shows the color variations determined by crop density (Normalized Difference Vegetation Index", or NDVI), where dark blues and greens indicate lush vegetation and reds show areas of bare soil.
- The middle image is a map of water deficit, derived from the Daedalus' reflectance and temperature measurements. Greens and blues indicate wet soil and reds are dry soil.
- The bottom image shows where crops are under serious stress, as is particularly the case in Fields 120 and 119 (indicated by red and yellow pixels). These fields were due to be irrigated the following day.

One of the spectacular advances in recent times is machine learning; applying computational methods to not only extract meaning from data, but to make optimal decisions based on this data. Recommendation engines suggest movies to us on Netflix and order search results on Google. Driverless cars are appearing on the roads. Can some of these technologies aid in improving food production and lead to meaningful response to global challenges of food security?

Our interest here is in algorithmically studying, and potentially optimizing, a number of operational aspects in certain parts of food production. We are interested in particular in hand-picked and 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 [3], (see also Figure 3). These crops can have very high value e.g. strawberry crops can yield $30,000 per acre [4], in contrast to row crops, at $8,000 per acre. Since mechanized platforms (e.g., harvesters) are 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 [5].

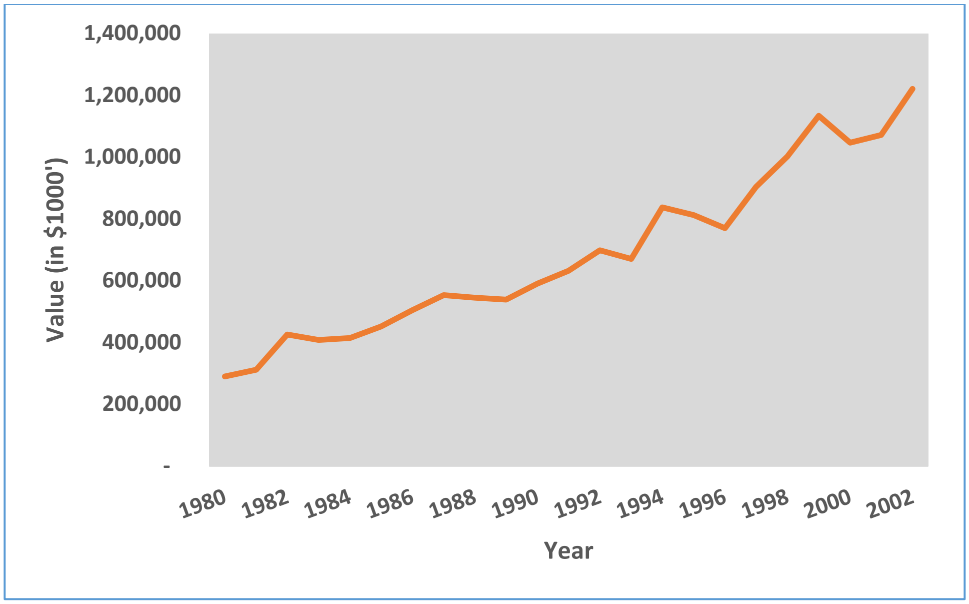


Figure 3 – Value of Strawberry harvested by Year

Labor is often the largest expense in the whole process (about 40% [3] for these crops; see also Figure 4). 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. What does precision agriculture mean in this case?

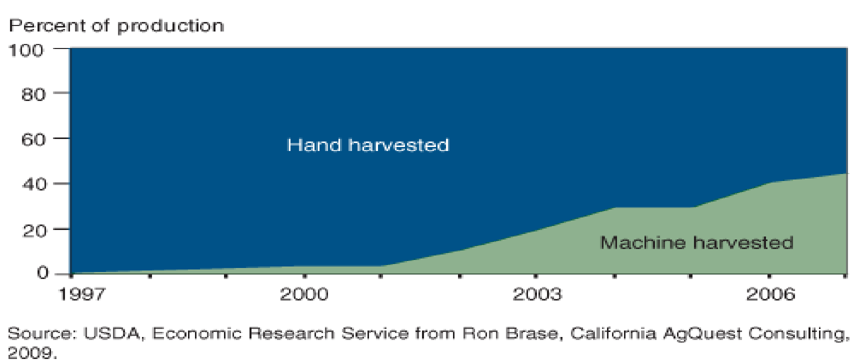


Figure 4 – Percentage mechanization and hand harvested man hours in raisin and strawberry farming

We have started working with a sample farm, Crisalida farms in Oxnard, California. On February 19th and 22nd of 2016, we visited and recorded GPS coordinates of 4 workers at 5-second intervals. The latitude and longitude tracks are in Figure 5 and 6.

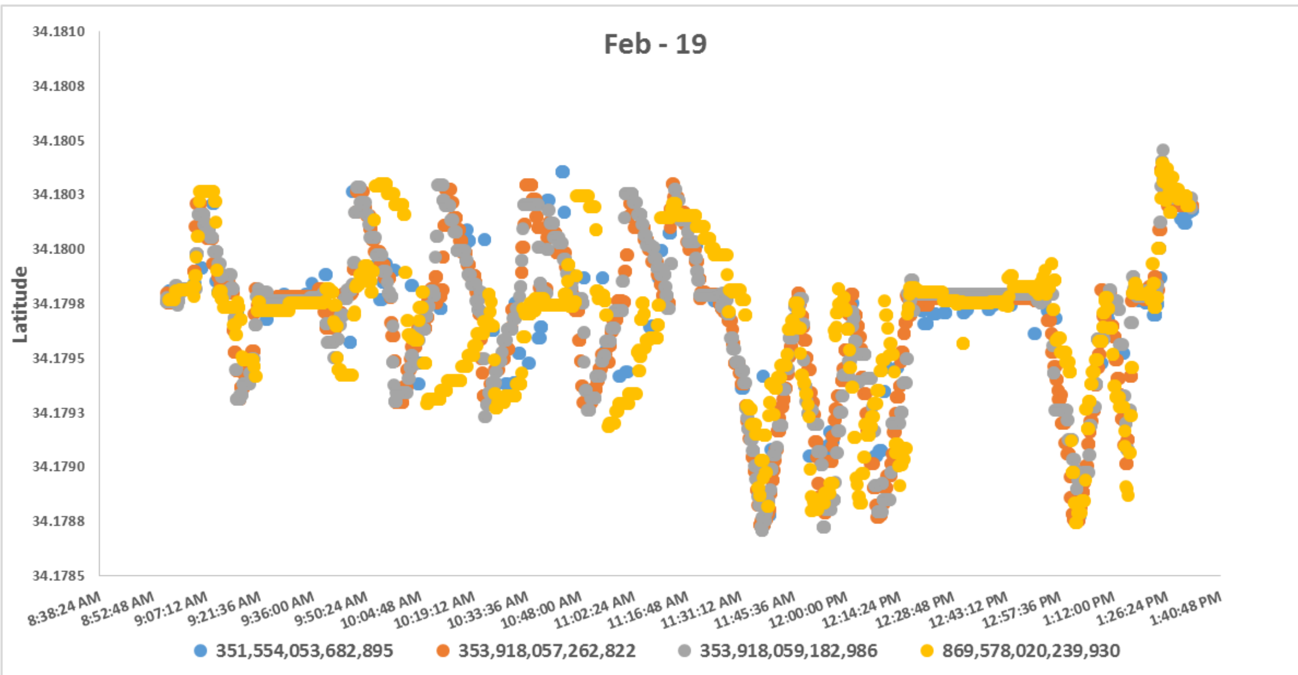


Figure 5 – Latitude trend against time

The focal point of the effort here is to algorithmically interpret geolocation data. In this case, we want to disambiguate several behaviors; i) harvesting, ii) lunch, iii) breaks. The larger issue for us is to quantify human labor as much as possible. Since human labor is extremely costly, we want to understand how to most efficiently use it.

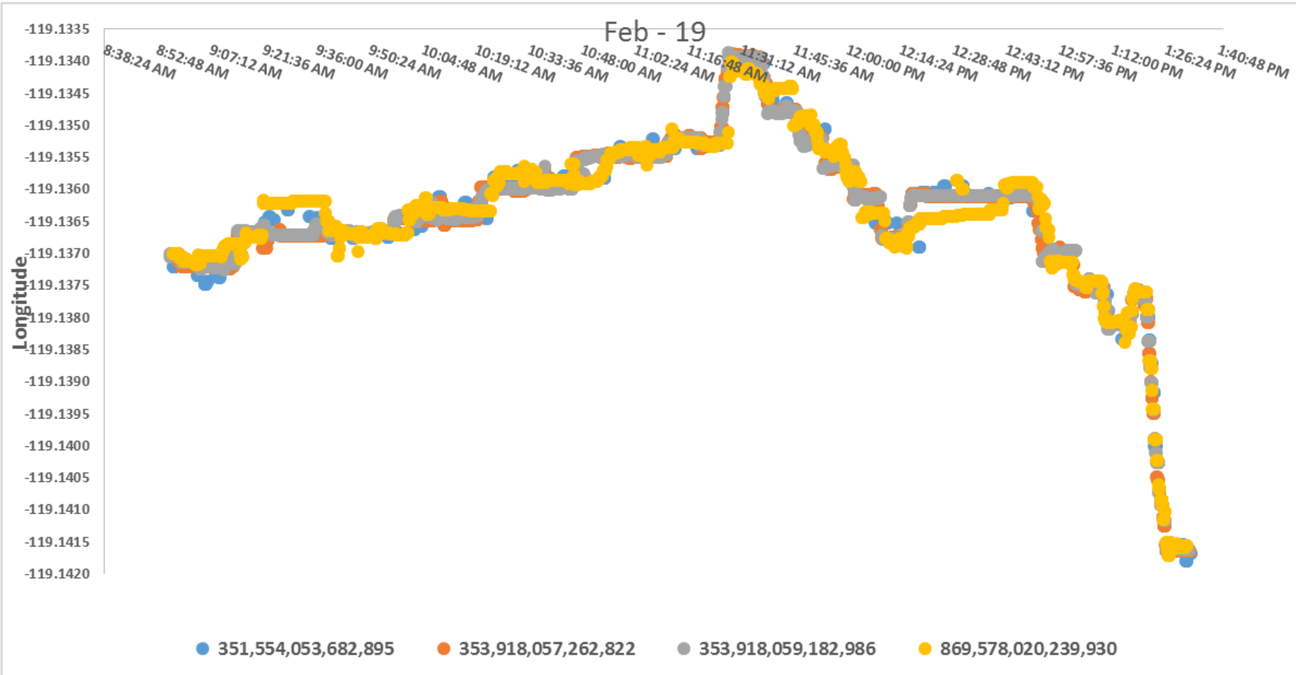


Figure 6 – Longitude trend against time

- How should it be deployed? How should a crew of laborer be optimally configured?

  Often the fastest worker sets the pace for the rest of the crew. However, if the fastest worker then ends up being idle, this is likely suboptimal. With real - time data, can labor be 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 (see Figure 7)?

Figure 7 – Cooling facility on a strawberry farm, such facilities are centrally located and simultaneously cater to produce from multiple locations

There are also other operational concerns surrounding labor. By law, harvesters must be given lunch breaks and warm-up time. Can data analytics provide sufficient verification of good labor practices? Harvesters are often paid piece-rate (i.e., according to amount harvested) and on an hourly basis. Since there are in fact various types of non-harvesting but necessary behavior (weeding, stacking of crates), identification of these different tasks may allow one to better assess pay schedules.

Literature Review

We look to the scientific community for guidance in pattern recognition using geospatial data, Google has an api – ActivityTracker[7] that is used in identifying different modes of activity be it rest, on foot(walking/running), on a bike, in a car by using the geospatial data of users from their phone. Our aim was to extend this functionality to agricultural activities while keeping the overall framework robust and as easy to scale as possible.

Our philosophy of cleaning data and then segmenting it into different segments is closely associated with that of Minhe, Chen and Zhang, 2008 [8], this helps limit the search space where we are looking for activities and it also helps increase the accuracy of detecting the optimal parameters associated with it.

Minhe et al hypothesize that there is a small pause between consecutive trips; different trips would often be intermediated by activities like walking to transition from one activity to the other. They then categorize any small rest and walking activities as transition points and group activity data on either side of the transition points as different activities to be identified. While the exact hypothesis does not hold true in our case, since many workers transition from one activity to the other without resting, we have imbibed their philosophy of dividing the entire dataset into contiguous data points segments each corresponding to a specific activity for ease of carrying out the computations.

Many different approaches have been tried for pattern recognition using GPS data, researchers in the past have used speed data derived from positional information to infer the transportation mode used by users at different times [9], we have used speed information in our segmenting trips in our case in a slightly different manner. We realized that GPS data was prone to inaccuracy of 5-15 m and hence calculating speeds from positional data would only compound the error, hence, we calculate moving average of 5 data points and fit piece-wise quadratic polynomials for every group. Parameters from the quadratic polynomial are used to estimate the radius of curvature and determine any steep changes in the profile of latitude or longitude data against time. This is used in turn to discern between different profiles.

This study was done with specific economic goals in mind – to curb the man hour losses due to inefficient vehicle scheduling. California experienced increase in minimum wage rates from $8/hour to $10/hour in 2016 [10]. Due to a shortage in farm help for harvesting, many farm owners have had to increase the wage to attract workers in harvest season [11]. Many farms have run out of business due to shortage of workers to pick their harvest which would soon rot if left in the sun [10]. Aforementioned factors have ensured that farm owners align their operations with profitability in mind and stem any possible losses.

While activity detection is the main focus of this problem, literature was also reviewed to explore the different techniques to generate forecasts from worker activity so that estimates of when a farm area would need a harvest transport vehicle can be created. Examples were explored where ARIMA models are used to fit a time series data either to better understand the underlying phenomenon or to predict future points in the series (forecasting). They are applied in some cases where data show evidence of non-stationary behavior, where an initial differencing step (corresponding to the "integrated" part of the model) can be applied to reduce the non-stationarity [12].

Data Collection

We visited Crisalida farms in Oxnard, CA in February of 2016. Four harvesters carried smartphones with an app which logged position and timestamp every 5 seconds. The smartphones were of various age and precision (Nexus 4's to Maven ZTE's). The latitude and longitude tracks of a single worker are shared in Figures 8 and 9.

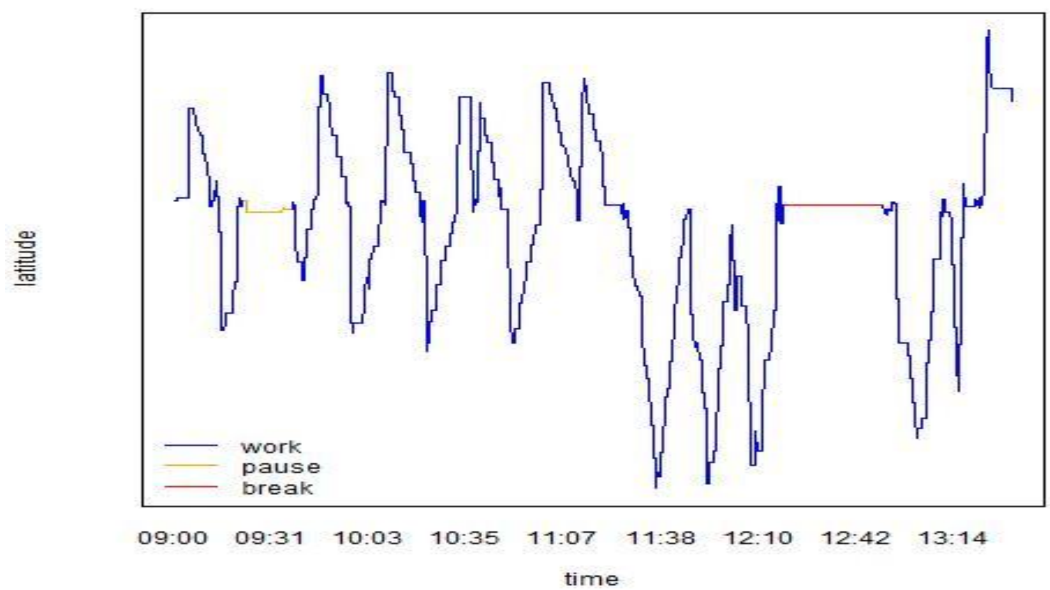


Figure 8 – Latitude information against time

We were also able to capture a battery charge indicator for the phones at every timestamp. This information could be crucial in determining the veracity of the positional data since phones are known to have trouble with GPS information while running on low battery. To record ground truth, we also made a note of all the activities performed by the 4 workers at different times during the day.

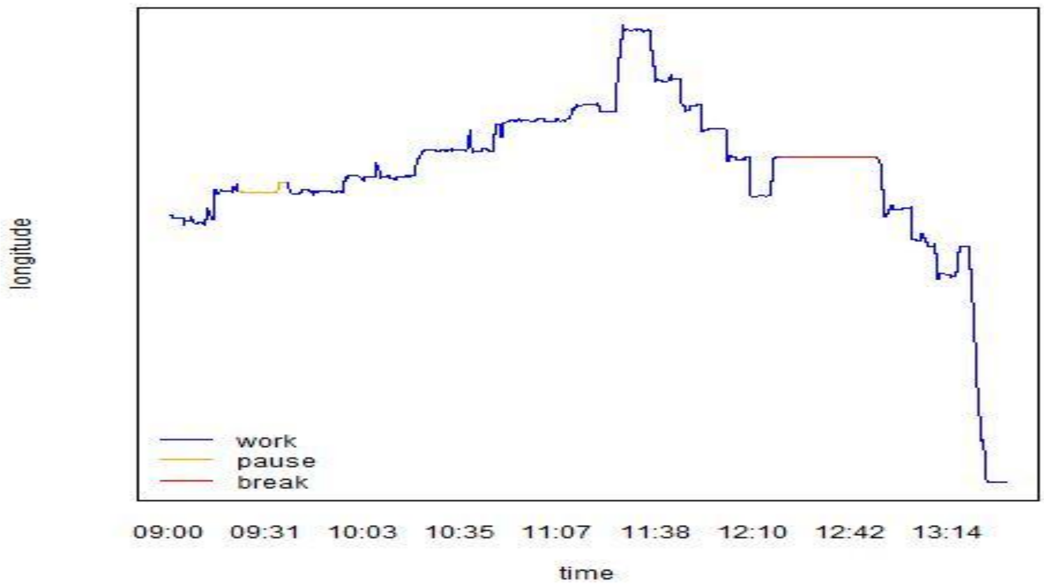


Figure 9 – Longitude information against time

This data was then collected using an android app – GPSLogger (see Figure 10); which would sends data to a mongo database maintained on Amazon Web Services servers.

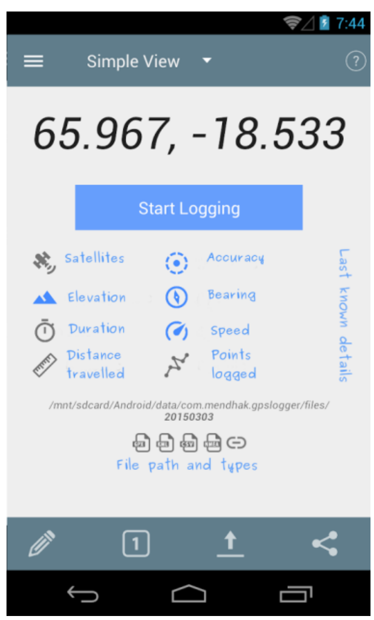


Figure 10 – GPS Logger App was used to log the Latitude and Longitude information to AWS

Methodology

A visual account was made of the farming activity by plotting the latitude and longitude information collected from the 4 phones on a regional map (see Figure 11), this was then tied with the positional trends for a worker – the idea being that if we could identify a consistent pattern in latitude (or longitude, or both) that corresponds to a specific activity performed by the workers (regardless of the worker under observation); then looking at the trends in the geospatial data, we can identify lunch, breaks and harvesting activity (where the harvesters walk the whole length of the field).

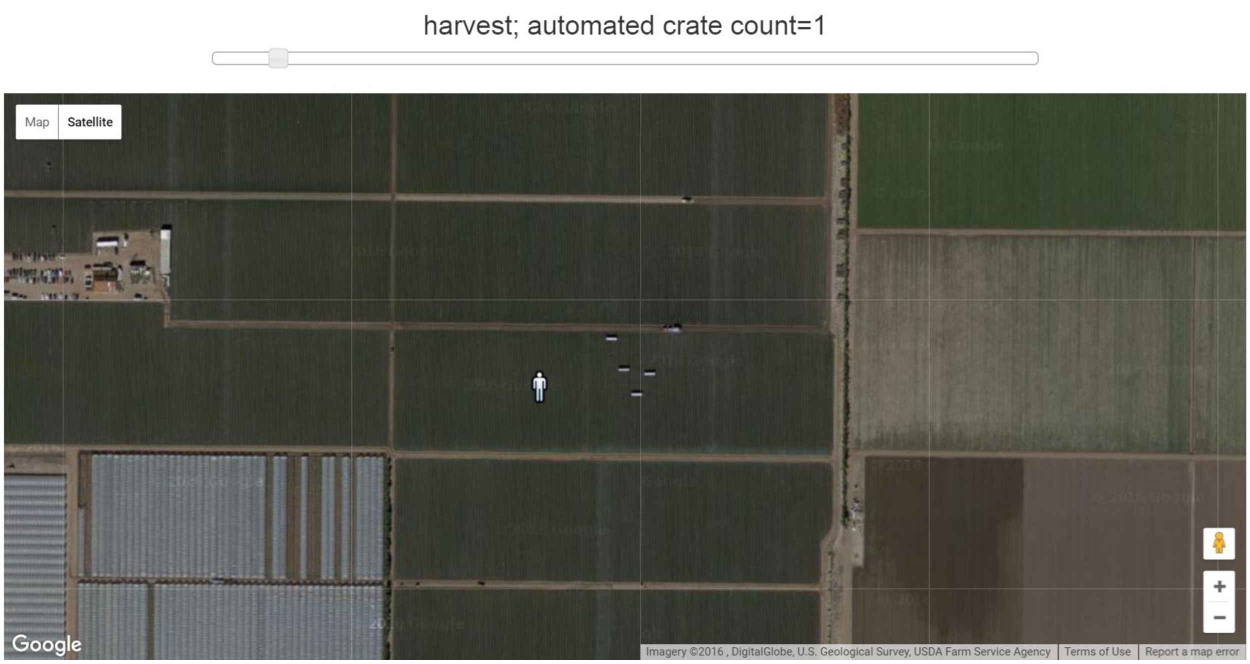
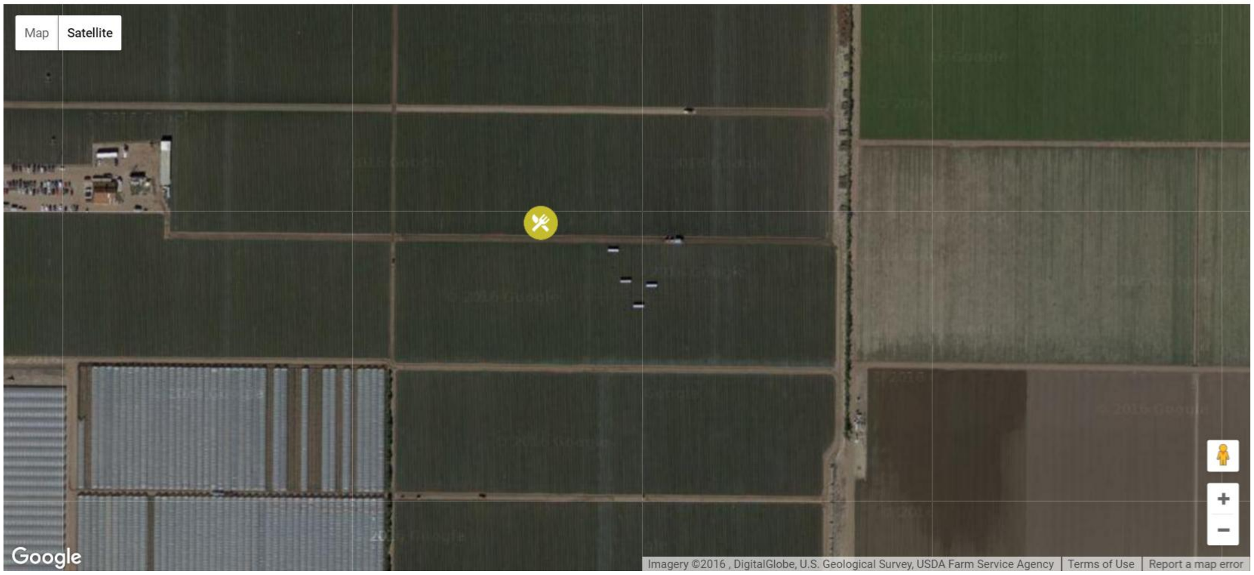


Figure 11 – Visualization of Positional Coordinates of a worker against time

break; automated crate count=1



weeding; automated crate count=4

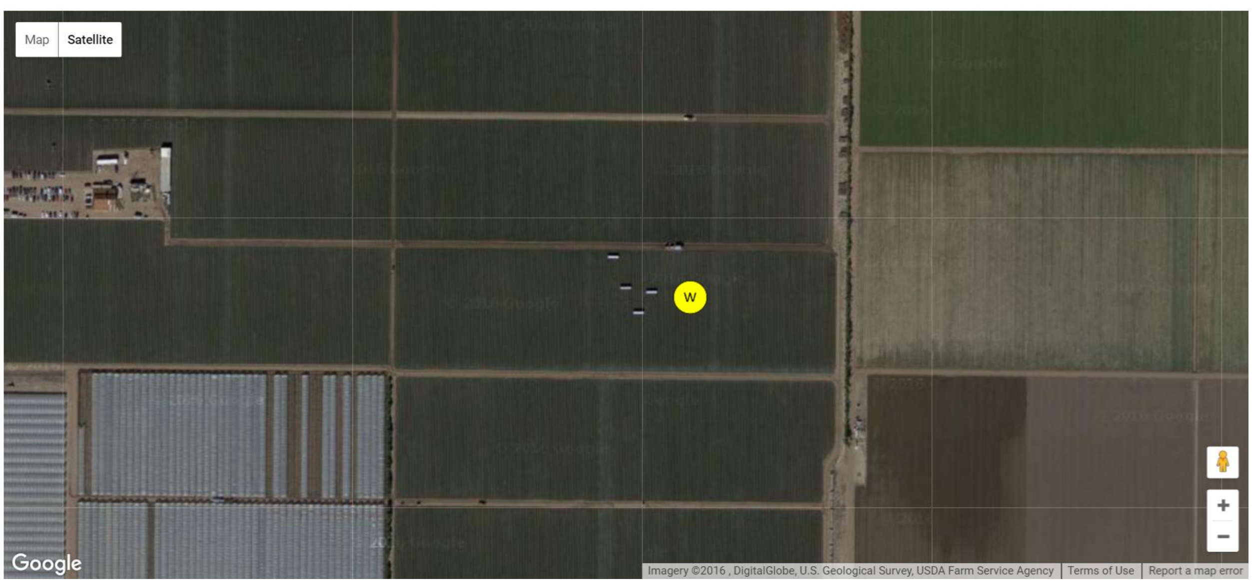


Figure 11 (cont.)

We note that in this case, the amount of data is initially fairly small; we don't have enough for a machine learning approach like clustering or classification. Hence, a pattern-matching approach was developed which, hopefully, will be enough to extract some information.

Let's look at a sample track of data from one worker; (see Figures 8 and 9, page 12). Fortunately, the rows in the field were 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.

In Figure 8, we see a number of excursions into the field, a break at around 9:30 in the morning, and a lunch break slightly after noon. We also see gradual longitudinal movement as row after row is harvested. We would like to disambiguate this behavior into several types of activity. We have several harvest patterns. Harvesting can occur either in a “hat" pattern, or in a “sawtooth" pattern. A break corresponds to no motion at all.

Let's define

$$h_H(t) \stackrel{\text{def}}{=} \begin{cases} t+1 & \text{if } -1 \le t < 0 \\ 1-t & \text{if } 0 \le t < 1 \\ \text{NaN} & \text{else} \end{cases}$$

$$h_S(t) \stackrel{\text{def}}{=} \begin{cases} t+2 & \text{if } -2 < t < -1 \\ -t & \text{if } -1 \le t < 1 \\ t-2 & \text{if } 1 \le t < 2 \\ \text{NaN} & \text{else} \end{cases}$$

$$h_B(t) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } -1 < t < 1 \\ \text{NaN} & \text{else} \end{cases}$$

Where h_H, a "hat" function is the signature of harvesting in one field, h_s, a "sawtooth" function, is the signature of harvesting across two fields, and h_B reflects a break (see Figure 12).

We want to decompose harvesting behavior into a "word" consisting of a sequence in h; e.g. H: = {$+h_H$, $-h_H$, $+h_S$, $-h_S$, h_B}. Outside of the domain of interest, we define the h_i's to numerically be NaN as setting them to be zero would overlap with the definition of a break.

In the case of our example (see Figure 8), we have the following sequence;

$$\boxed{H: = \{h_S, h_B, h_S, h_S, h_S, h_H, h_H, -h_H, -h_H, -h_H, h_B, -h_H, -h_H\}}$$

These patterns represent departure from common latitude, and have varying widths, heights, and translations. Furthermore, there are some points which don't fall into any one of these behaviors. We also note that for our example, H is a vector of length – 13.

We define domain or support for a profile as the number of points for which the profile returns a non-NAN output. We also define a cost function to evaluate the fit of different profiles for a set of data points; each profile can take different values for each of the parameters; namely - translation, height and width as input. Next, we optimize over all the profiles for a given data set and choose the one with the lowest cost, the corresponding parameters are chosen as “optimal” parameters.

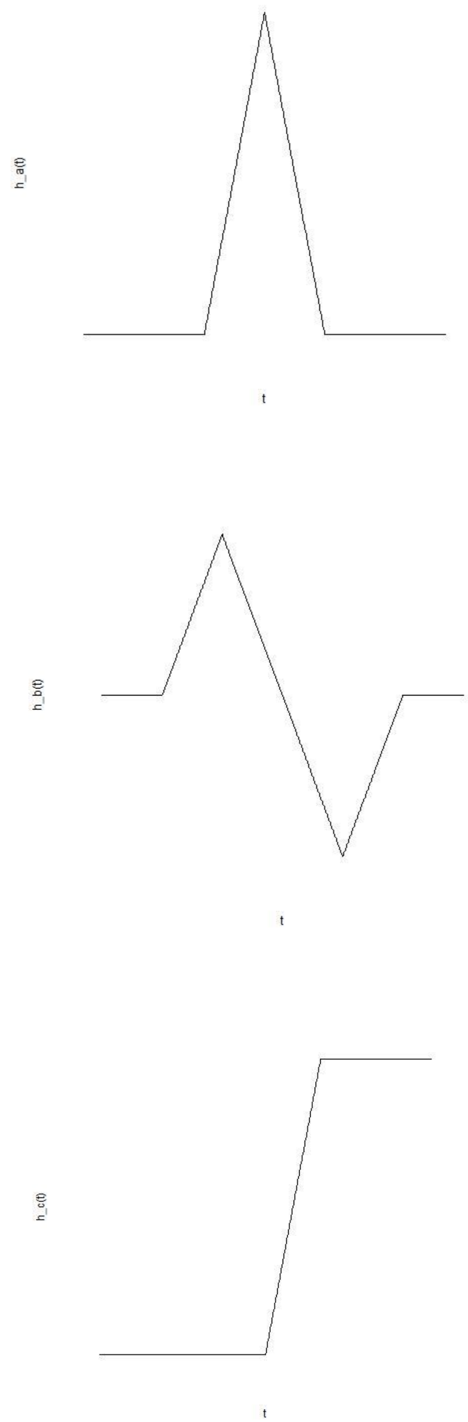


Figure 12 – Hat, Sawtooth and Weeding profiles (top to bottom)

Formally, we define the cost J as –

$$J_H^{(1)}(\bar{a},\bar{\beta},\bar{\gamma},c) \stackrel{\text{def}}{=} \frac{1}{N_1}\sum_{i=1}^{13}\sum_{n=1}^{N_1}\left|y_n^{(1)} - \beta_i h_i(\gamma_i(t_n^{(1)} - a_i)) - c\right|^2 \mathbf{1}_{\{\gamma_i(t_n^{(1)} - a_i))\in D(h_i)\}}$$

The vectors $\bar{a} \in R_+^{13}$, $\bar{\beta} \in R^{13}$, $\bar{\gamma} \in R^{13}$ and $c \in R$ parameterize transformations from the reference patterns to the observed patterns; $\bar{a}$, $\bar{\beta}$, and $\bar{\gamma}$ represent, respectively, the horizontal translated, width, and height of the pattern h_i which we are trying to compare to the dataset, and c represents the reference latitude.

Since we have already divided the entire dataset into segments of data points each representing a particular activity, we are able to provide strict bounds to the optimization problem of finding the optimal activity within the different dataset segments. This is done to save time finding the optimal activity by providing limits to the search space in which the optimal parameters exist.

We use the L-BFGS-B method to conduct the optimization; this is a modification of the popular Newton's method for optimizing a multivariate scalar function. Unlike the Newton's method which requires the computation of the Hessian matrix for the function for every iteration; the L-BFGS-B method creates an approximation to the Hessian that only has to be updated. Another optimization applied to the method is that instead of storing the entire matrix, this method stores just the vectors that represent the approximate Hessian matrix implicitly hence saving on memory requirements required across iterations.

We also add on a penalty which discourages configurations which leave a large number of data points un-approximated (i.e., points in the data set that are not in the Domain of any of the h_i); let's define penalty P as –

$$\tilde{P}^{(1)}(\bar{a}, \bar{\beta}, \bar{\gamma}, c) \stackrel{\mathrm{def}}{=} \left| \left\{ n \in \{1, 2 \ldots N\} : \gamma_i(t_n^{(1)} - a_i) \notin \cup_{i=1}^{13} D(h_i) \right\} \right|$$

The augmented cost function J becomes

$$\tilde{J}_H^{(1)}(\bar{a}, \bar{\beta}, \bar{\gamma}, c) \stackrel{\mathrm{def}}{=} J_H(\bar{a}, \bar{\beta}, \bar{\gamma}, c) + \alpha \ln \tilde{P}^{(1)}(\bar{a}, \bar{\beta}, \bar{\gamma}, c)$$

Here α is a penalty constant. In practice we can carry out this procedure simultaneously for all the workers. The cost function changes with penalty constant (see Figure 13).

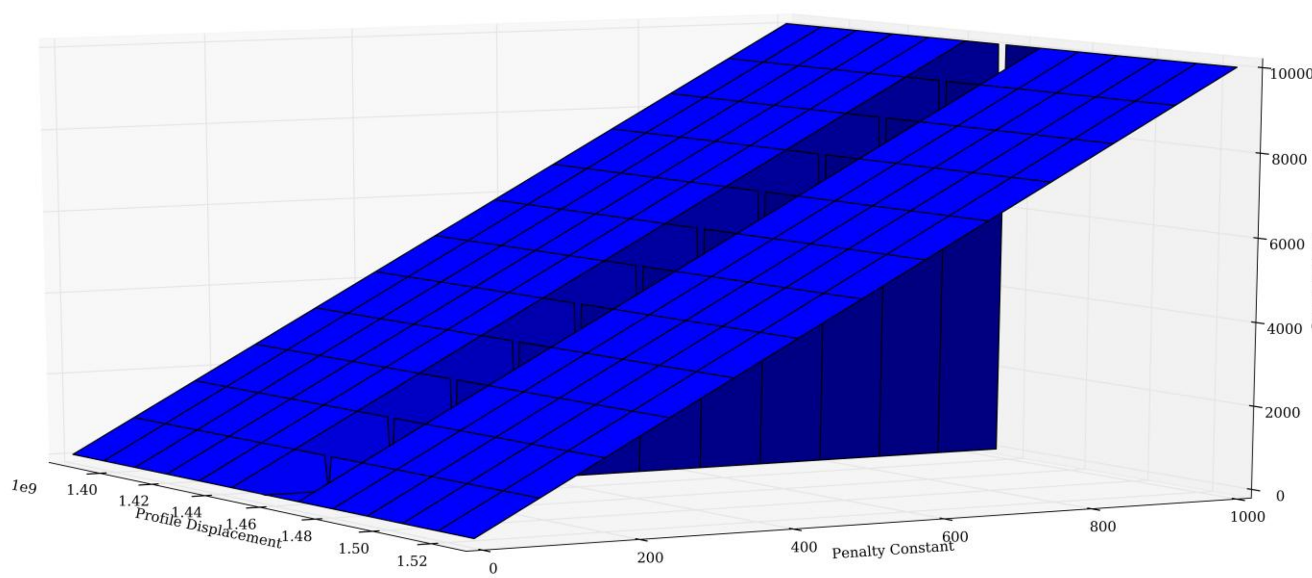


Figure 13 – Cost function against Penalty Constant and Profile Displacement

We have used a linear penalty function; the cost function increases in proportion with the penalty constant.

Results

The activities tagged by the algorithm for different workers for Feb – 19 and 22 are summarized in the table below. The results from the model are compared against observations made on the field.

A worker's entire day of activities as detected by the code are summarized in the figure(s) below

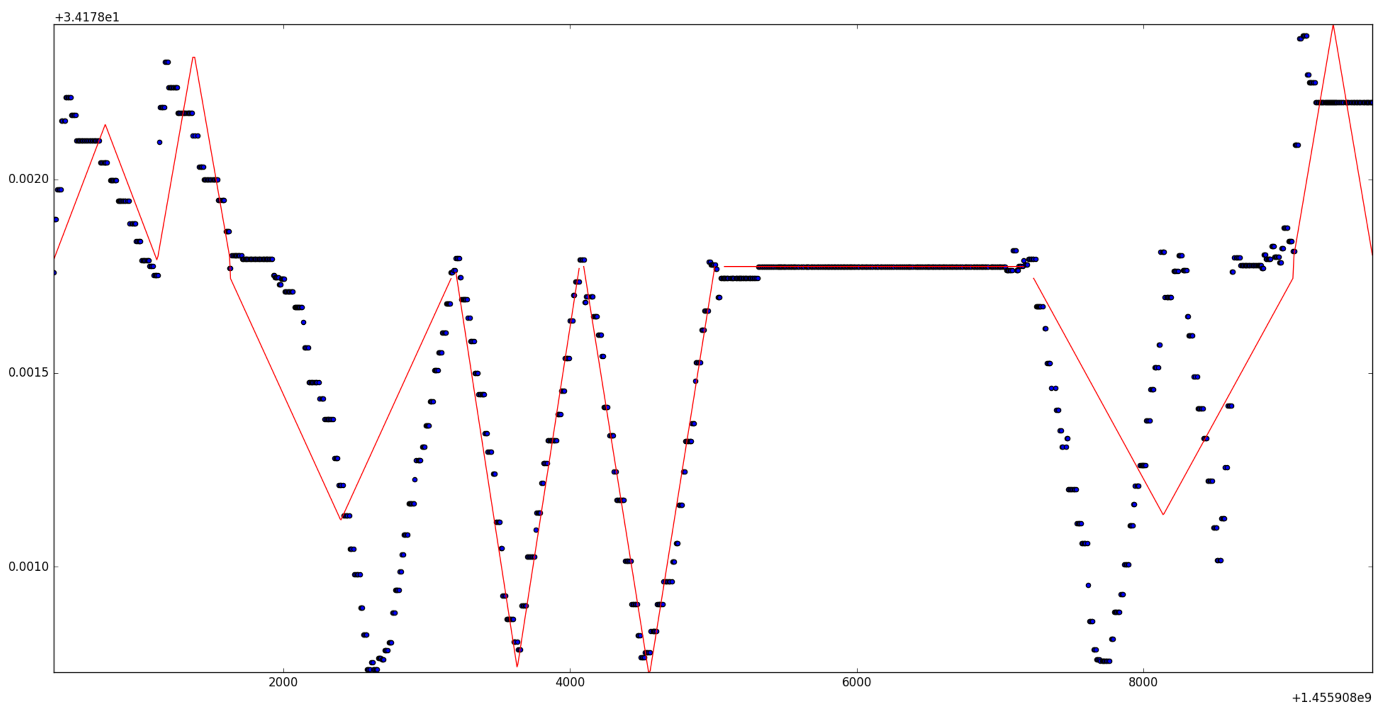
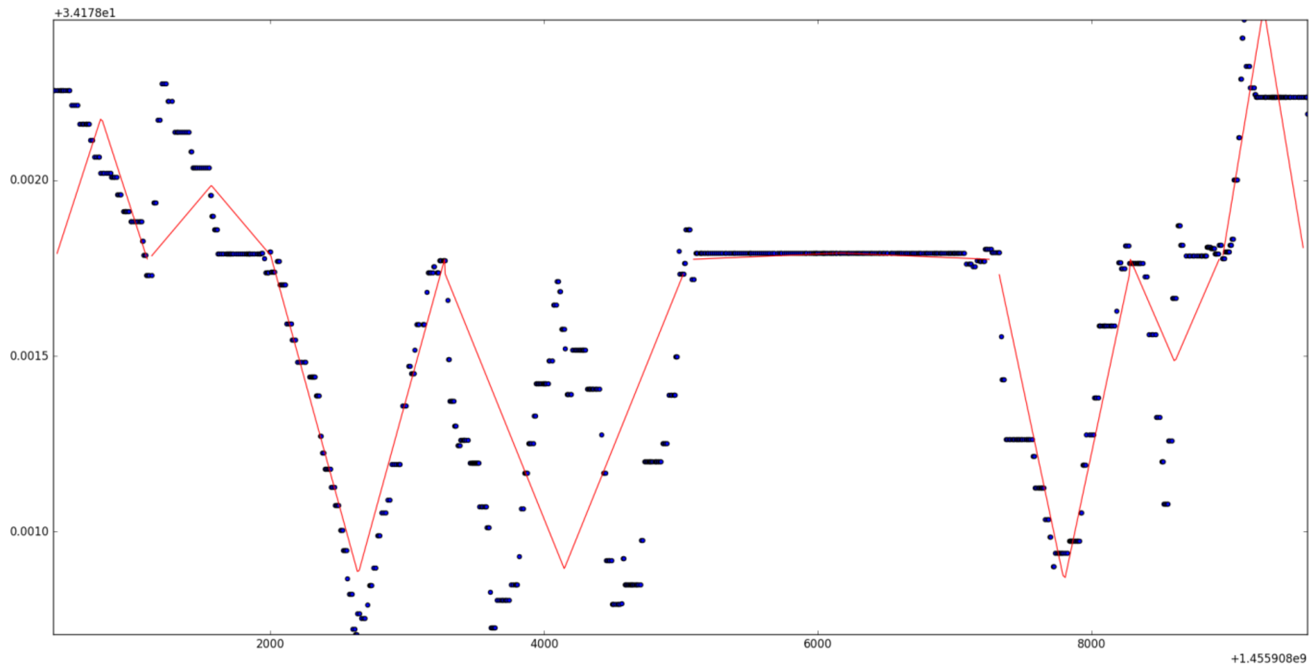


Figure 14 – Afternoon data for Workers (1 on top, descending downwards) for 2/19.

The profiles detected are marked in red, it is worth noting that due to insufficient data for Worker 4 (graph on the bottom) the profiles detected do not match the data patterns.


+3.4178e1
0.0020
0.0015
0.0010
2000
4000
6000
8000
+1.455908e9

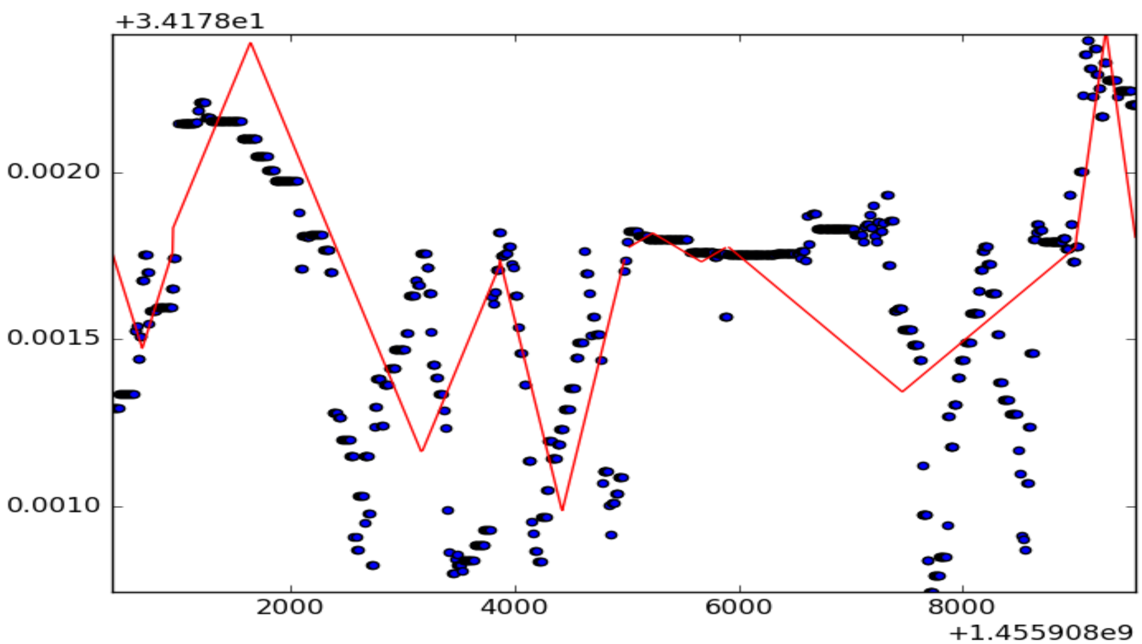

+3.4178e1
0.0020
0.0015
0.0010
2000
4000
6000
8000
+1.455908e9

Fig 14 (cont.)

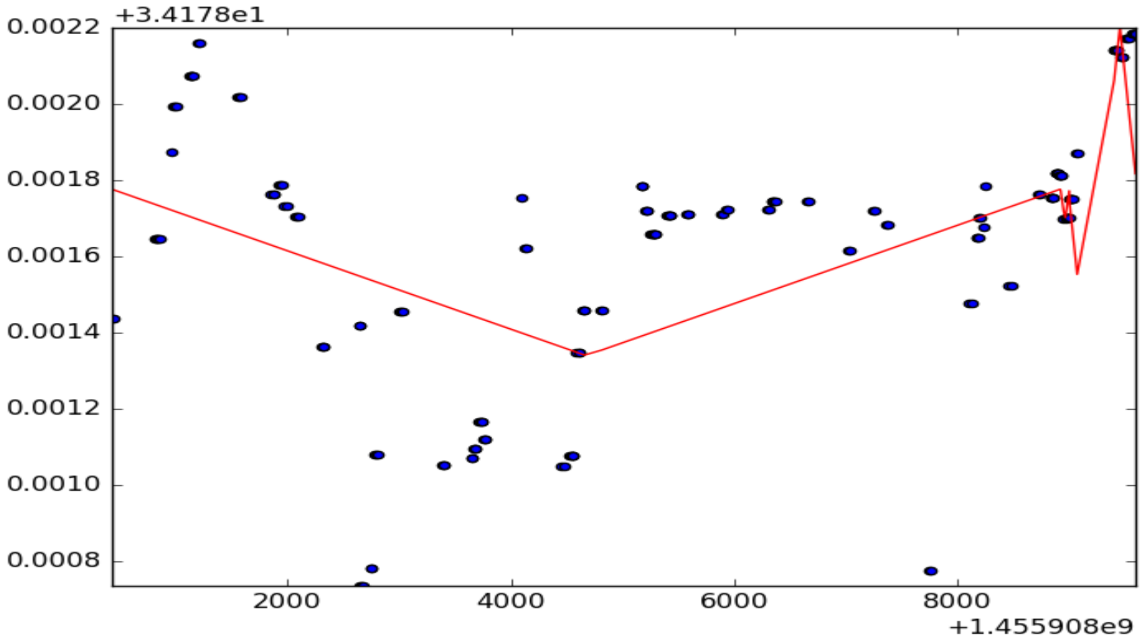


Figure 14 (cont.)

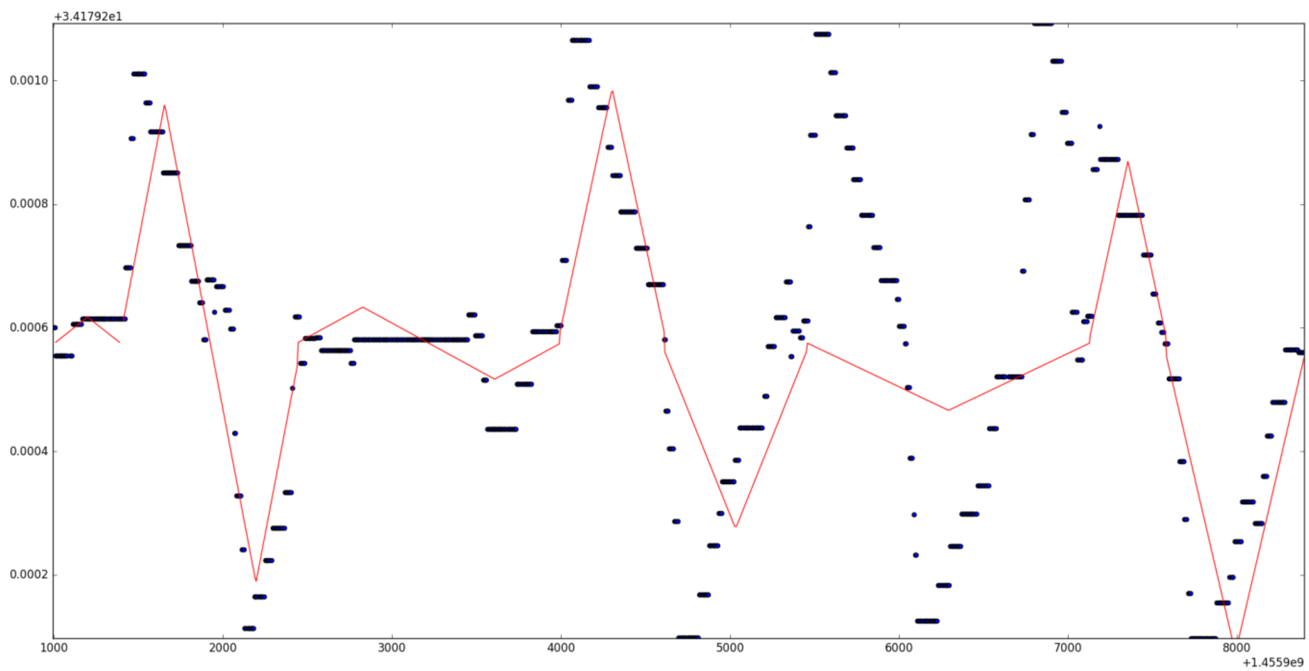
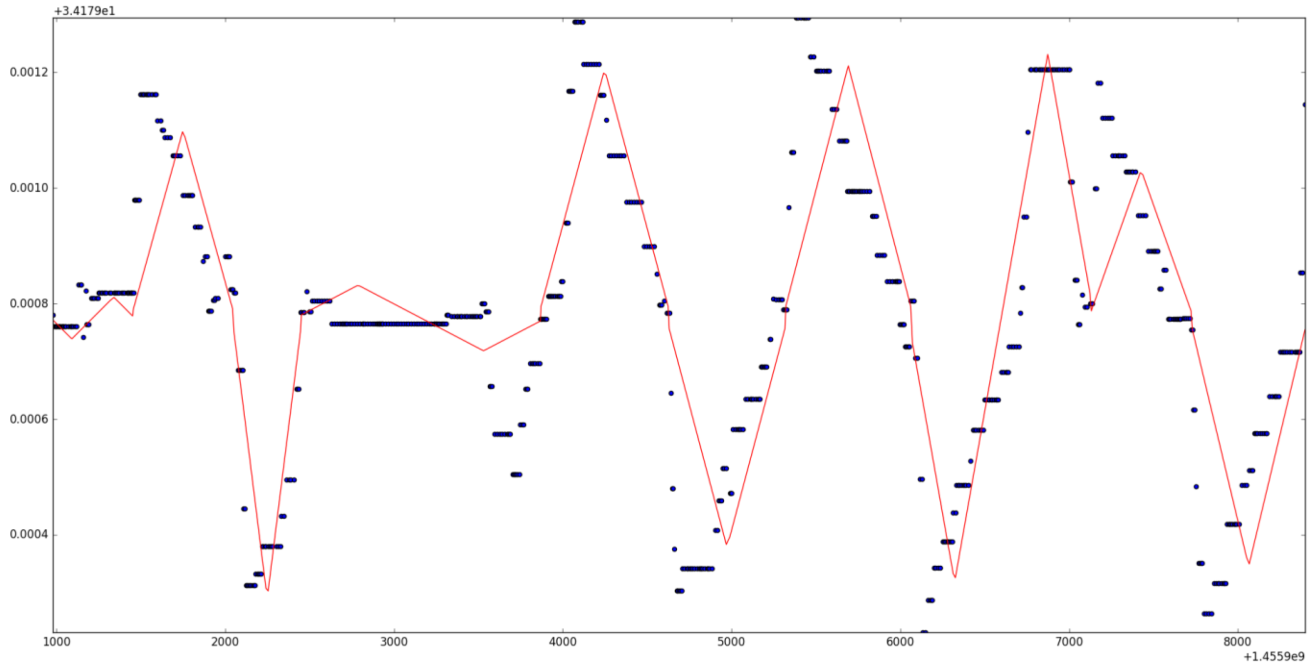


Figure 15 – Morning data for Workers (1 on top, descending downwards) for 2/19.



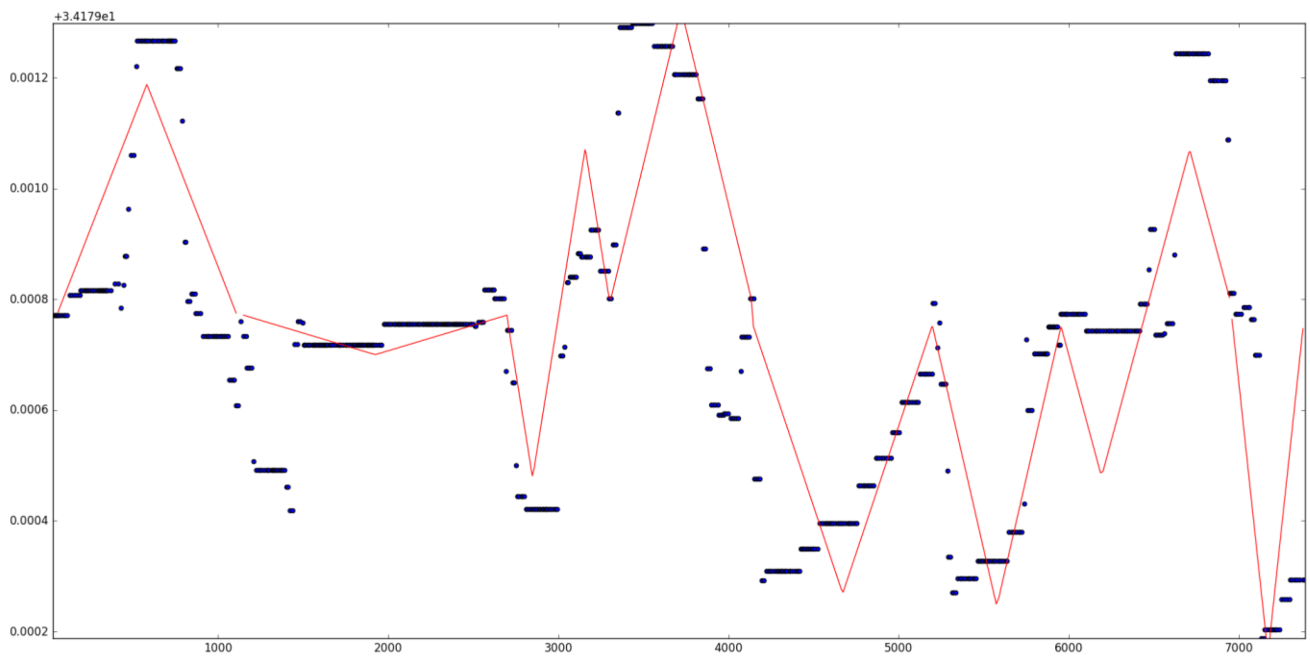


Figure 15 (cont.)

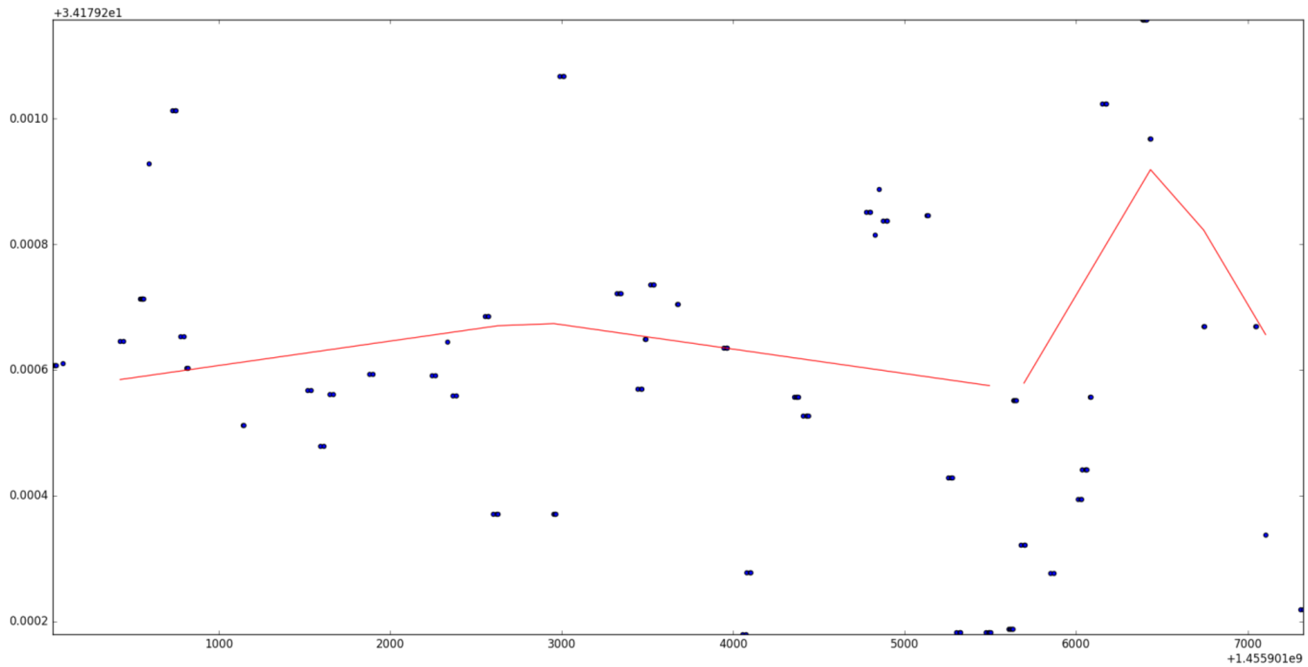


Fig 15 (cont.)

The profiles detected are marked in red; it is worth noting that due to insufficient data for Worker 4 (graph on the bottom) the profiles detected do not match the data patterns.

The runtime of the code on a worker's entire day of activities are summarized in the table below (see table 1)

Day	Worker	Shift(A/M)	NumPoints	Runtime(s)
2/19/2016	0	A	1294	43.02
2/19/2016	1	A	959	38.19
2/19/2016	2	A	974	32.85
2/19/2016	3	A	187	13.73
2/19/2016	0	M	1051	39.54
2/19/2016	1	M	741	40.66
2/19/2016	2	M	770	36.20
2/19/2016	3	M	138	10.54

Table 1 – Runtimes of code for data of 19th Feb, 2016

Challenges

There were a number of challenges faced during the completion of the project. Some were due to the nature of the problem, while the others arose due to scalability required in code execution. In this section we attempt to highlight some of such challenges and the steps that can be taken to meet them.

1. Basin of attraction for parameters – There was evidence of shallow basin of attraction for parameters in the cost function (see Figure 16), i.e. the cost function would not change appreciably when parameter values lie outside the neighborhood of optimal values while changing drastically when the parameter would lie within the optimal neighborhood. This meant that if the initial values for the parameters were not chosen within the optimal neighborhood, the parameters returned from optimization would be sub-optimal, in some cases the optimization scheme would not converge. As a solution, the cost function was evaluated at all the points of a fine bounded mesh of parameters. This however increased the runtime of the code considerably since the number of computations increased by a factor of the mesh size.

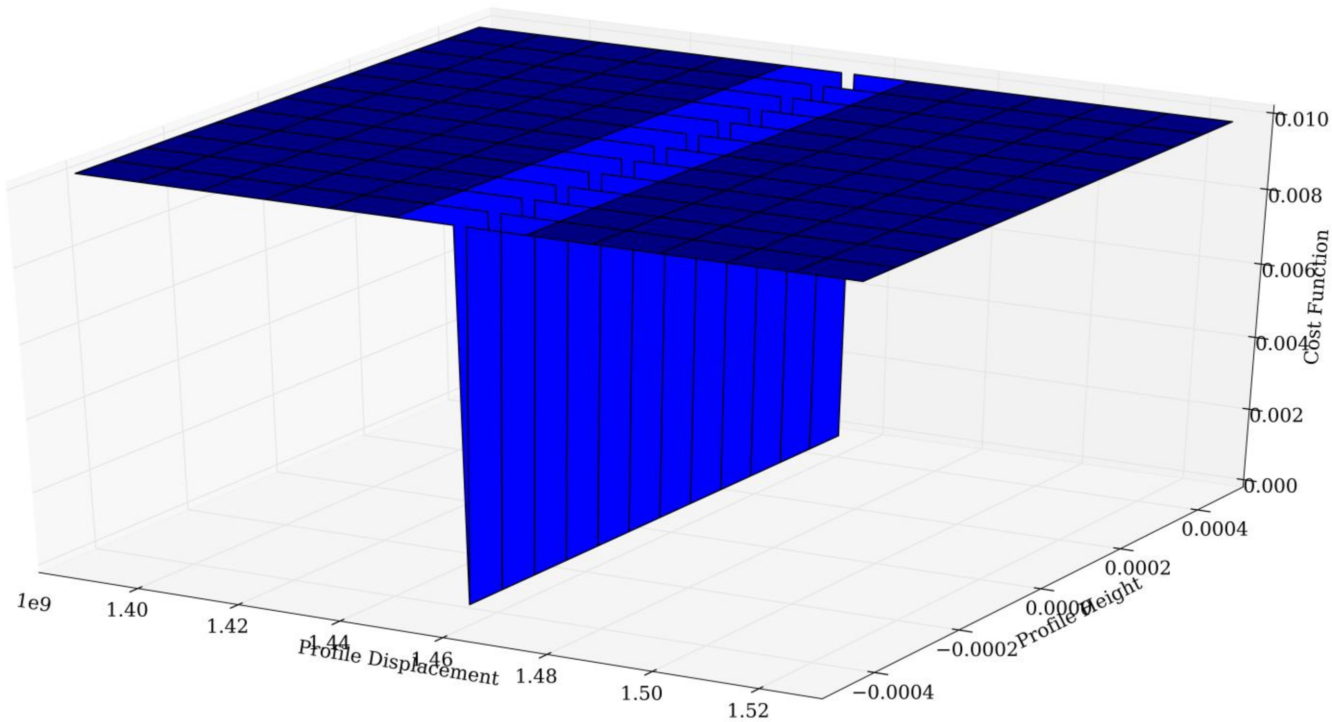


Figure 16 – Cost function against Profile Displacement and Height

We notice the deep valley in the cost function around the optimal value of profile displacement.

We also notice that the cost does not change appreciably for different values of height.

2. High Computation time for small mesh sizes (11 per parameter) – The cost function was evaluated at different points of a bounded mesh to determine the optimal parameter values for a profile, there was, a computational overhead associated with each evaluation which led to an increase in the code runtime. This problem can be circumvented by making use of the instruction level parallelization. GPU cores can be programmed to use thread blocks and execute code simultaneously.

3. Veracity of the data points – Positional data used in the project was collected using phones; these phones belonged to different generations and had different implementations of GPS technology in-build. As a result, there was disparity in the accuracy of the positional coordinates recorded from the phones. The battery of the phone also affected the accuracy of the data points. To solve this issue – we have not looked at single data-points; rather, we look at a collection of 5 data points at the same time to detect a change in profile since the probability that 5 consecutive data points will be equally affected by a random error is miniscule. Also, while looking for an optimal profile for a data segment we look at the entire segment together since the probability that the all the points have errors which makes them "seem" like a different profile (e.g. Break looks like a hat) is relatively low.

4. Surveying the actual area to get baseline and plot boundary data points – This project primarily depends upon the positional data from workers. There are however, some measurements that may need an onsite-survey by a PI. We may also require a reference recording of the activities performed by a sample of workers at different times so as to calibrate the findings of our model.

Conclusion

The aim of this study was to investigate the possibility of activity disambiguation on a farm land using just the basic geolocation coordinates against time. We utilize the principle of "temporal cohesion" and "spatial cohesion" and hypothesize that data-points of a person collected around the same time will be located closely, and vice versa. We then create rolling groups from this data (size 5 presently, but can change) and locate different times when a worker's mode of activity changes (e.g. the activity changes from "Break" to "Hat"). We then group all the data points under a particular activity together and determine the optimum profile and its parameters.

This study aims, in philosophy, to extend the ActivityRecognition api offered by Google. The idea in this case is to use the location data of any user and identify if they are farming, taking a break or weeding out the furrows for future use. The study does have some shortcomings that need to be worked upon before this project can be scaled up to a large number of users. Current code runtime is high and it makes the app unamenable to use at scale – instruction level parallelism needs to explored to get the runtime down, PyOpenCL offers a great opportunity to do so.

While this study offers a good step forward into the potentials of using geospatial data in precision agriculture there still remains a lot of work to be done to use this code and generate $ saving recommendations from the data (e.g. work remains to create a vehicle routing algorithm or a forecast algorithm for the farm)

An interesting application of the study can also be in integrating the worker profile with the farm land to schedule workers intelligently to different parts of the farm that best matches their individual strengths and weakness; e.g. a person with high endurance could be sent to cover a large piece of farm with moderate yield while a person with higher focus could be sent to a smaller piece of farm with high yield.

In conclusion, we would like to summarize that the results obtained from the study have proved that there is a lot of scope for application of data analytics into the field of agriculture. The benefits from the burgeoning field of data analytics shall affect large and small farm land owners alike and so it is imperative that these applications reach the end decision makers.

Proposed Plan for the Future

As a proof of concept this study was designed to be easy to implement and maintain financially viability at the same time, it leverages the all-pervasive android technology and comes up with a framework that is real-time and requires very little setting up. Use of technologies like the Amazon web services makes the project cost-efficient since it means that the hardware associated with the project can be hired on a pay per use basis.

Crisalida farm was generous enough to provide us access to their workers for the purpose of data collection, for the next phase, this project is to be rolled out across all the workers employed by the farm. We would test and deploy the code to cater to an increase in the number of workers by ~x1000; the scale of such an operation would mean that we would have to model (or potentially re-model) our data structures to be able to derive performance from the code at scale.

Under the ambit of this project, we have attempted to isolate points where a worker changes their mode of working, and identify the activities they undertake from time to time. Further to this process, we would aim to build a real-time forecast to estimate time of completion for all the parts of the farm; this would give the farm owners a good sense of how the farm is performing – holistically and in parts. This could be consumed as a real-time report with relevant kpi's to keep a check on the performance of the farm and make sure all the functions run harmoniously.

Man-hours lost due to lack of empty crates at different plots contributes significantly to losses incurred by the farm, in addition, strawberries tend to rot in the sun when they are not transported to the cooling facility soon after harvest. These losses can be directly attributed to mismatch in scheduling the transport vehicles to farms, therefore, upon creation of a forecast for different plots in the farm, we would create a scheduling algorithm to use the real-time forecasts created of the different plots to allocate transport vehicles so that workers always have access to empty crates for collecting their harvest and the strawberries do not rot in the sun.

The algorithm designed to detect profiles in the data is amenable to parallel programming; the current code takes ~2 minutes to detect different profiles and find the optimal parameters for 4-hour activity data of 4 workers. In its current state the code will not be able to provide real-time updates to worker activities as the number of workers increase. We can write code to take advantage of computing power of GPU's available in most modern computing devices to significantly reduce code execution time.

OpenCL provides the framework required to write and execute programs across heterogeneous platforms containing CPU's, GPU's etc. PyOpenCL is a wrapper written in back-end C that brings out all the functionalities of core OpenCL to python, the same can be installed on top of OpenCL to program GPU's in python. PyOpenCL can be installed on Amazon Web Service machines to make use of the GPU's installed for conducting computations.

References

[1] World population projected to reach 9.7 billion by 2050, Department of Economic and Social Affairs, UN, July 2015 - http://www.un.org/en/development/desa/news/population/2015-report.html

[2] Global agriculture towards 2050 - High Level Expert Forum, October 2009 - www.fao.org/fileadmin/templates/wsfs/docs/Issues_papers/HLEF2050_Global_Agriculture.pdf

[3] Strawberries in the US - Econ 496, Iowa State University, December 2004 - http://www2.econ.iastate.edu/classes/econ496/lence/spring2004/strawberries.pdf

[4] Facing the Future in a Global Economy - The U.S. Produce Industry and Labor, Calvin et al. November 2010 - http://www.ers.usda.gov/media/135123/err106.pdf

[5] Piscadores De Fresas, Santa Maria Ca, Youtube video cited on July, 2016 - https://www.youtube.com/watch?v=CFORJwDMxqs

[6] Farm economy, US Department of Agriculture, Economic Research Service, July 2016 - http://www.ers.usda.gov/topics/farm-economy/farm-labor/background.aspx#Numbers

[7] Google Activity Recognition Api - https://developers.google.com/android/reference/com/google/android/gms/location/ActivityRecognitionApi

[8] Minhe Ji, Chen and Zhang, FKSD 2010 - "Identifying Travel Mode from GPS Trajectories through Fuzzy Pattern Recognition"

[9] Huss, Beekhuzien, Kromhout and Vermuelen, PMC 2014 - "Using GPS-derived speed patterns for recognition of transport modes in adults", PMCID: PMC4320483

[10] California minimum wage increase, Janas 2015 - https://www.wga.com/blog/2016-california-minimum-wage-increase

[11] Marysville, California Farms Are Desperate for Workers, Despite High Unemployment – Huffington Post, May 2012 - http://goo.gl/SxlSEV

[12] Asteriou, Dimitros; Hall, Stephen G. (2011). "ARIMA Models and the Box–Jenkins Methodology". Applied Econometrics (Second ed.). Palgrave MacMillan