Land Cover Change Detection using Spatio-temporal Vegetation Data

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Yashu Chamber

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Vipin Kumar

August, 2012

Acknowledgements

I am thankful to my advisor, Prof. Vipin Kumar, for his continued guidance during my masters degree at the University of Minnesota. His support and feedbacks have been invaluable towards the successful completion of my thesis as well as throughout my academic stay at the university. I also want to thank research associates in my team as well as my colleagues for their help and advice during my graduate studies.

Dedication

To those who matter

Abstract

Vegetation change, especially in forests, is one of the key contributors to atmospheric carbon imbalance. Detecting and quantifying changes in vegetation is helpful for various purposes, including climate prediction modeling, global warming studies as well as reporting and tracking illegal deforestation. There are broadly three types of vegetation changes: sudden change (e.g. due to fires), land cover type change (e.g. forest conversion to farmlands), and gradual change (e.g. due to logging, insect infestation, reforestation). Gradual change is one of the most prevalent type of vegetation change in the world. In this thesis, two methods for detecting vegetation changes from spatio-temporal datasets are described. The first is a time series based approach that focuses on gradual changes while the second is a spatio-temporal approach focusing on both gradual and sudden changes.

Contents

List of Tables

List of Figures

Chapter 1

Introduction & Related Work

1.1 Introduction

The vegetation on land forms a skin on Earth's surface, and forests in particular are one of earth's most valuable assets. Forests are critical to all life forms, home to all wildlife and one of the key moderators of the global climate system. Preservation and protection of forests is therefore vital to the overall health of the planet. But reduction in forests take place in various forms both naturally, as forest fires and insect infestations, as well as through human activities like logging (for purposes such as commercial use, urbanization, conversion to farm lands and illegal trading). At the same time, planting trees tend to mitigate the destructive effects of forest loss and help restore balance. However still, forest reduction may cause permanent loss in fauna which may never be revived through reforestation. For all of these reasons, detecting and quantifying changes in forests, and vegetation in general, is critical for efforts to preserve the forests. Furthermore, quantification is also useful for related purposes including climate prediction modeling, global warming studies as well as reporting and tracking illegal deforestation.

There are broadly three types of vegetation changes: sudden change (e.g. due to fires), land cover type change (e.g. due to forest conversion to farmlands), and gradual change (e.g. due to logging, insect infestation, or reforestation). Remote sensing offers rich data sets that are very well-suited for monitoring vegetation around the globe, in a regular fashion across time. A large variety of techniques and tools have been

developed for detecting changes in forest cover, and more generally land cover [1, 2]. Gradual change, however, has not received much attention as compared to the sudden changes [3], but nonetheless, it is very important because it can occur due to a variety of causes as indicated above. In addition, the cumulative effect of gradual change can have large impact on carbon balance, microclimate, and biodiversity patterns [3]. However, detecting gradual forest degradation (as opposed to abrupt changes caused by fires etc.) is particularly challenging because the reduction in forest cover occurs very slowly and the amount of reduction observed across time is small compared to natural variations.

In this thesis, we present two approaches for detecting vegetation change. PART I describes a time-series based method for detecting gradual changes in vegetation from spatio-temporal datasets [4]. The approach is robust, scalable and easy to apply across different regions and vegetation types. PART II describes a spatio-temporal approach to detect persistent vegetation changes.

1.2 Related Work

Traditional approaches that use image based comparison for detecting forest degradation are frequently domain-specific or region-specific [5] which often require expensive training, and are thus not suited for application at global scale. More recently, time series based methods applied on remote sensing datasets have gained much attention to detect deforestation because of their scalability, accuracy, and forest monitoring capability at frequent intervals. However, even most of the current time series based approaches for detecting vegetation loss in forests are aimed at only certain types of changes (e.g. due to fires), which are characterized by sudden and severe vegetation loss [6].

A number of approaches have been proposed for identifying gradual changes in a time series. [7] describe the use of the well-known CUSUM technique for land cover change detection. CUSUM follows a simple approach of determining deviation in the values of a time series from an expected value, and the change score, giving the magnitude of change, is determined as the maximum cumulative deviation. Another approach presented recently by [8], Breaks for Additive Seasonal and Trend (BFAST), decomposes a time series into trend, seasonal and residual components. The time series is divided into segments such that intra-segment trend is constant, while inter-segment trends are

dissimilar. A trend breakpoint is associated with segment boundaries. The seasonal component is handled in a similar fashion.

[3] proposed another approach for gradual change detection based on single segment trend analysis. It studied gradual vegetation changes in four regions in the United States: forests and rangelands in the southwestern US, sagebrush in Wyoming, woodland in Nebraska and White Mountains in New Hampshire. They analyzed trends in linear regression models of time series of two different vegetation indices, Normalized Difference Vegetation Index (NDVI) and Short Wave Infrared / Near Infrared Index (SWIR/NIR), constructed from Landsat Thematic Mapper (TM) and Enhanced Thematic Mapper Plus (ETM) reflectance bands. Depending on vegetation type, SWIR/NIR was the main vegetation index for analyzing the conifer forest trends in southwestern US and the White Mountain of New Hampshire, while NDVI was the primary vegetation index in Wyoming, Nebraska and New Hampshire. The slope of regression lines were analyzed at different statistical significance levels and spatial trends were observed at certain p-values. In addition, depending on the region, ancillary data were used to interpret the trends.

Yet another land cover change (forest disturbance) detection approach, LandTrendr [9], employs a regression-based temporal segmentation approach to capture trends in Landsat time series data. The overall aim of this approach is to transform the given time series into a sequence of trends which forms the trajectory of the given time series. The *first step* is to identify the maximally complex model of the time series given two control parameters, $max\_segments$ and $vertexcountovershoot$. In this step, after the time series is despiked, the time series segmentation begins by considering the entire time series as a single segment with two initial vertices: the first (first vertex) and the last (second vertex) value of the time series. To get the third vertex, a first-order regression is performed on time series values and the third vertex is identified as that time series value that is farthest from the regression line. Thus two segments are obtained, one by connecting the first and the third vertex, and the other by connecting the third vertex and the second vertex. The segment with the higher mean squared error (MSE) is selected to repeat the above process and identify another vertex (and thus another segment). This process is repeated till the segment count reaches $max\_segments + vertexcountovershoot$. The excess segments are culled by removing the vertex with

the shallowest angle and updating the segmentation with the new set of vertices. This process is repeated till the segment count reduces to $max\_segments$. After obtaining the segmentation, the trajectory-fitting algorithms (point-to-point or regression-based fitting) are employed to adjust the spectral value of each vertex which would result in "best continuous trajectory" through the time series. The *second step* is to simplify the model by removing the weakest vertex (defined by MSE or recovery rate) and reapplying the trajectory-fitting algorithm. This is repeated to generate segmentation models for segment count ranging from $max\_segments$ down to one. The *third step* is to select the best model based on simple fitting statistic and the *final step* is to remove change that is considered noise based on a vegetative cover criterion.

Chapter 2

Data Description

The time series data set used for this study is the Enhanced Vegetation Index (EVI), which is a product based on measurements taken from MODIS instrument on NASA's Terra satellite, and is available for download from the Land Processes Distributed Active Archive Center (LP-DAAC) [10]. EVI measures the "greenness" signal and essentially reflects the amount of vegetation at a location. The spatial resolution of the dataset is 250 meters or 1000 meters and the temporal resolution is 16 days (23 time steps per year) covering the time period from February 2000 to the present. The range of EVI is 0 to 1, where 0 indicates no vegetation and 1 indicates vegetation saturation. Figure 2.1 shows an example of an EVI time series.

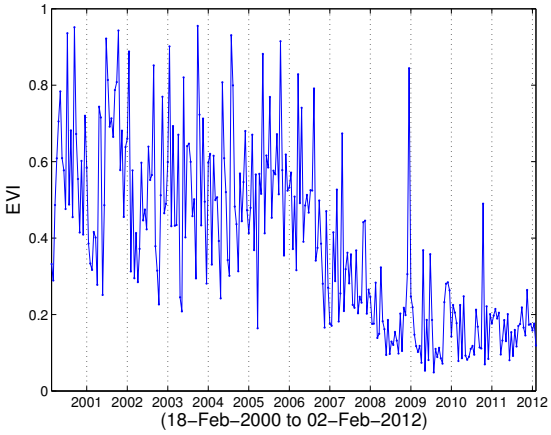


Figure 2.1: Example of an EVI time series (with noise in observations).

Remote sensing data sets are frequently subject to contamination due to clouds, haze, pixel geometry and other factors. We preprocess the EVI time series data set in order to remove undesired fluctuations in EVI (such as the sharp increases in Figure 2.1). This improves the effectiveness of identifying signatures of interest. For smoothing

purposes, we have used the Savitzky-Golay smoothing filter [11], which uses two parameters: polynomial *degree* desired for smoothing and *frame size*. The smoothing filter fits a polynomial function of the indicated degree over a window equal to the frame size over each time step, the current time step being at the center of the window; the EVI value of the current time step is then replaced with the polynomial fit.

Part I

Time Series based Approach - Persistent Delta (PDELTA)

Chapters 3 - 7

Chapter 3

Problem Formulation & Proposed Method

3.1 Problem Formulation

Given an EVI time series dataset, we are interested in identifying time series such as the one shown in Figure 3.1, where there is a perceptible decrease in the signal, along with determining the approximate period of decrease (as shown by the vertical lines in Figure 3.1). Identifying a time series with a gradual decrease in vegetation is challenging due to a number of reasons: distinguishing vegetation loss from natural seasonal variations; differentiating between a spurious decrease due to noise or environmental factors and a genuine decrease caused by degradation on ground; correctly determining the period of decrease (start and end time steps) especially when there is a high degree of variability in the time series. There could also be a phenological change during the decrease period or across the decrease, and the algorithm must be able to handle such cases and extract the decrease period appropriately.

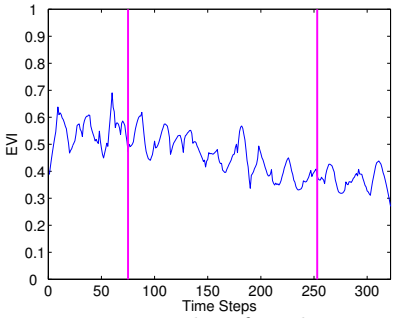


Figure 3.1: Example of a decreasing time series. Vertical lines enclose a gradually decreasing segment.

n	The number of time steps in a time series.
S	The number of time steps corresponding to one year of data (we also call this the season length). For biweekly data used, $S = 23$.
t_1	First time step.
t_i	ith time step.
v_{t_i}	Data value at the time step t_i
$v_{t_i}...v_{t_j}$	All values between time steps t_i and t_j.
T	A sample time series $= v_1 v_2 v_3 ... v_i ... v_n$
$v_{t_i...t_j}$	$mean(v_{t_i}...v_{t_j})$
Δ_i	$v_{t_{i-S+1}...t_i} - v_{t_{i+1}...t_{i+S}}$
Δ-series	$\Delta_S \Delta_{S+1} \Delta_{S+2} ... \Delta_{n-S}$

Table 3.1: Notation for time series change detection.

3.2 Proposed Method Background

A reduction in forest cover is often reflected as a decrease in the EVI value. In fact, many existing schemes compute difference in EVI (or related indices) between different years to identify changes. However, the values of vegetation indices such as EVI can have a high degree of variability due to seasonality (e.g. vegetation is greener in the summer than in the winter in the temperate zones), as well as due to natural variation in vegetation growth caused by environmental factors such as temperature and precipitation.

Most existing methods have handled seasonality by comparing vegetation index values at (or around) the same date in different years. Natural variability is much harder to handle since it is difficult to characterize. The problem becomes even more acute for many non-forest covers such as shrubs, since natural variability tends to be much larger in these cases. Although our focus is on identification of degradation in forests, it is not possible to completely exclude non-forests from any study due to the unavailability of highly precise forest maps [12].

3.2.1 CUSUM Method for detecting decreasing time series

CUSUM, proposed by [13], is a well-known change detection algorithm originally developed in the domain of process control. The basic CUSUM scheme has an expected value μ for the process represented by a time series. A cumulative sum, CS, of the deviations of values of the series from μ is maintained. If there is no change in the process, the values of CS series is expected to be low; if CS exceeds a user-defined threshold at any time step, the time series is flagged as changed.

There are multiple ways in which CUSUM can be used to assign a change score to a time series, the simplest of which is to use $\max\{CS_1, CS_2, \ldots, CS_n\}$. However, this score can be sensitive to noise [14]. [7] developed a CUSUM approach for gradual and sudden land cover change detection which uses a more robust technique to compute the change score. Specifically, a bootstrap procedure is used to determine the confidence of CS by determining the degree to which such a score can occur by chance. The bootstrap procedure involves randomly permuting the input time series to obtain a distribution of change scores $\mathcal{R}$ (CUSUM is run on each randomization). The confidence of the drop is determined by the relative frequency of CS being greater than the randomized distribution, i.e. $\frac{|CS>\mathcal{R}|}{|\mathcal{R}|}$.

[7] take the expected value μ as the mean of the entire time series. Other measures may also be used to compute μ such as the value of the first time step, or the mean of the first S values. The advantage of using the mean value across a periodic cycle over a single time step is that the mean value is independent of the fluctuations in a time period (or seasonal variation in case of the MODIS EVI time series).

Some drawbacks of the scoring mechanism of CUSUM described above are: (1) *Change point of drop or the period of decrease not identified.* This method only identifies a score corresponding to the maximum deviation in cumulative sum time series, and does not give the period of change. (2) *Computed score may not be associated with the decreasing period.* Computed score is the maximum cumulative deviation from the expected value, which may or may not depict the amount of EVI lost during the decrease period.

3.2.2 Adapting CUSUM for Gradual Degradation

In the original CUSUM approach, the expected value is always fixed in a time series regardless of the way it is computed. In this study, we propose a different strategy: If we take the expected value at any time step t_{i+1} as the value at time step t_i, then the deviation of values at each time step from its expected value would give the amount of drop or rise from its previous value. However, such a model would be dependent on the intra-periodic variation and the resulting deviation could be due to the natural periodicity of the time series. In order to make this process independent of periodicity, averaging over a periodic cycle can be used. Therefore, instead we take the mean value of the current periodic cycle as the expected mean value for the next periodic cycle. The deviation between mean values of successive periodic cycles is also equivalent to the drop in EVI across a time step t_i that marks the boundary between these periodic cycles, i.e. the one that ends at t_i, and the other that begins at t_{i+1}. We refer to this type of differencing (computing drop from previous periodic cycle in succession) as *Successive Differencing* (SD), which is different from computing the deviation from a fixed expected value as done in CUSUM, which we refer to as *Fixed Differencing* (FD).

If T is a given time series, n is the number of time steps in T, and S is the periodicity of T, we can define SD and FD methods as:

FD: $\Delta_i^c = v_{t_i} - \mu \quad \forall \quad i \in 1 \cdots n$

SD: $\Delta_i^s = \Delta_i \quad \forall \quad i \in S \cdots n - S \qquad (Refer \quad to \quad Table\ 3.1 \quad for \quad notation)$

For MODIS EVI time series, Δ_i^s's are computed as difference between the mean values of two successive years (two consecutive sets of 23 values since $S = 23$).

FD gives deviation in values relative to the fixed expected value, while SD gives the drop relative to the previous year. Also, in the first equation, individual data values are used for differencing while in the second equation, averaged value over a seasonal cycle is being subtracted. Computing drop from a previous value in succession can provide trend information in a time series, which is what successive differencing does. Also, subtracting averaged values instead of individual values make the trend information more robust to seasonal variations and noisy outliers. On the other hand, Δ_i^c's fluctuates with the seasonal variations, even when there is no decrease in vegetation. Also, it does not provide trend information which is vital for identifying decreasing periods in a time

series.

3.2.3 Using Successive Differencing

As we have seen above, successive differencing using mean value of annual segments can be used to determine trends in a time series. Therefore, we use SD instead of FD for our approach. Below, we mention some possibilities in which SD could be used:

Consider a method for detecting gradual decrease that tries to identify the window in a time series that has the largest drop: given a time series, identify two years, i.e. two sets of 23 consecutive time steps, y_1 and y_2, such that the difference between the mean EVI of y_1 and y_2 is maximum in the time series. This is similar to identifying the window where the sum of Δ_i^s is maximum. This method will work well for consecutively decreasing time series. However, there are some disadvantages of this method when applied to a time series with high variation. The primary disadvantage is that this method loses information about the time steps in between y_1 and y_2. For example, given the time series shown in Figure 3.2, this method would identify the decrease as having occurred between years 2 and 11 even though it is clear that the time series increased significantly after year 7. Specifically, if the time series rises and falls in between then such a time series is highly variable and it should not be considered as changed. Another disadvantage is that if there are large spikes in a year due to noise that distorts the mean EVI for that year in an otherwise stable time series, this time series will be given a high score (drop from y_1 to y_2) by this method, even though this change is spurious.

To overcome drawbacks of the above method, yet another method to detect gradual decrease could be to compute the difference between successive yearly sets (Δ_i^s), and determine the longest continuous window of positive Δ_i^s. This method again has a major disadvantage that if there is a spurious rise in time series due to noise, the drop window will fall short of that false rise and thus could be determined much smaller than it actually is (e.g. for the time series in Figure 3.2, this method would incorrectly detect end of degradation in year 5).

Building on the concepts described above, we propose a novel time series change detection method, *Persistent*-Δ, or $PDELTA$. It uses successive differencing as the base to compute Δ_i's. The key property of successive differencing is that as long as

there is a decrease in the time series from one year to the next, Δ_i would be positive. If the decrease is at an almost constant rate, the Δ_i would be almost constant. As soon as Δ_i becomes zero, it means that there has been no vegetation change from past year to the present year. But it could be too soon to say that the change in vegetation has stopped since this could be due to some noisy time steps and it's possible that after very few time steps, Δ_i's become positive and stay positive for a few years. Thus the change didn't really stop, but continued after a short time. Therefore, the primary objective of PDELTA is to determine the window of *maximum reliable drop*. This method tolerates natural variation which may cause small increases in individual years during an extended period of degradation. For example, in Figure 3.2, the technique correctly identifies the period between years 2 and 6 when degradation has occurred since it accounts for the perturbations in the intervening years. However, if a rise in the time series violates a *reliability condition*, the approach differentiates it from the natural variation and does not consider this in the changed phase. The next section describes the PDELTA method in detail.

3.3 Method - PDELTA approach to detect change window

As mentioned in the previous section, the main objective of the PDELTA approach is to identify the window of maximum reliable drop in a time series. It need not be a continuous drop, and some amount of intermittent rise can be allowed as long as a decreasing trend is persistent. The amount of intermittent rise allowed is controlled by a *Reliability Condition* which limits the amount of every intermittent rise during an extended period of degradation. The remainder of this section describes the approach in detail.

As a first step, we compute Δ_i, as described in the previous section, at each time step beginning at the end of the first year (since we don't have sufficient information to compute Δ_i during the first year) and terminating before the start of the last year (again due to insufficient information during the last year). Let the series composed of Δ_i's, $S \leq i \leq n - S$, be Δ-series (Delta-series) where S are the number of time steps in a year, and n are the number of time steps in the time series. Next, we identify those time steps in the time series that have the characteristic to become the extremes of

the drop window. For this effect, we compute a Γ-series (Gamma-series) from Δ-series, with each time step represented by γ_i, using the following transformation:

$$\gamma_i = \begin{cases} 1 & \Delta_i > 0 \\ -1 & \Delta_i \leq 0 \end{cases}$$

We say that those ith time steps are candidates for drop start for which γ_{i-1} is -1 and γ_i is 1 (transition to 1). Similarly, those ith time steps are candidates for drop end for which γ_{i-1} is 1 and γ_i is -1 (transition from 1). The bottom plot in Figure 3.2 show an example of the Δ-series, Γ-series scaled by a factor of 0.07, and candidate start and end time steps (b_1, e_1, etc.).

Let there be K candidate start and stop time steps identified in time series T, which are denoted by b_k and e_k respectively, $\forall k \in 1 \ldots K$. Between every b_k and e_k there is a decrease in EVI values (decreasing trend), and between every e_k and b_{k+1} there is an increase in EVI values (increasing trend). For each b_k we are interested in identifying the farthest e_l ($1 \leq k \leq l \leq K$) such that the time series pattern within these limits in general has a decreasing trend, even if there are mild rises in between. If a drop starts at b_k, then an intermediate rise occurs between every e_l and b_{l+1} ($k \leq l < K$). In order to ensure that the decreasing trend is followed across these intermediate rises, we test for a drop reliability condition at every e_l which must be satisfied before allowing the rise between e_l and b_{l+1}. This reliability condition is given below.

Reliability Condition (RC): It states that after the commencement of a drop at a time step b_k, the rise occurring at a certain time step e_l ($e_k \leq e_l < e_K$) would not be considered as drop termination if this rise is bounded by a fraction (x%) of the drop that has already occurred, and, if there is an *overall decrease* in the EVI values during the time steps in the *limited future* of e_l. The *limited future* is defined as the time steps following e_l during which the rise does not exceed x%. The motivation for this is that if the current drop window has accumulated a large sum of Δ_i, a greater room for rise is allowed as long as the time series in general still has a decreasing trend.

As long as this condition is satisfied, e_l can be extended. As soon as this condition fails at a certain candidate stop time step, we stop at that e_l, and the maximum reliable drop that began at time step b_k is terminated at e_l. This becomes a candidate drop

window (cw_p) for time series T. Since there could be more such windows in the same time series that begin at other candidate start time steps, b_j, $j \neq k$, we repeat the above process for the remaining candidate start time steps. Thus, we would have at most K candidate drop windows cw_p, $\forall p \in 1 \cdots K$.

We compute a score for each candidate window using one of the methods described in Chapter 4. The maximum scoring window is determined as the representative window of the time series. Thus, the significance of change in each time series is given by the score computed for their corresponding representative change window. The higher the value, the more severe the change. Currently, we identify a single representative window of a time series because we are interested in identifying time series that have undergone change at least once. However, this is not a requirement and multiple segments with significant decreases can be selected from the same time series - multiple change detection.

The example time series in Figure 3.2 captures the effectiveness of this approach. The top plot shows the EVI time series, the smoothed time series, and the two vertical solid lines identifying the drop period. It can be noticed that the time series gradually decreased over many years and then stabilized. The bottom plot shows the corresponding Δ-series in solid curved line and the Γ-series scaled by a factor of 0.07 by a broken line. Notice that the first drop in Δ-series below zero is included in the maximum reliable window since the reliability condition is not violated.

Let us also consider a case of a spurious rise as shown in Figure 3.3. In such cases the drop window resulting from this rise will be small since the subsequent period will not be able to satisfy the reliability condition. Furthermore, the score of the identified drop window according to our methodology (as described in Chapter 4) would be low. Hence these type of drops would be easily differentiated from the genuine drops.

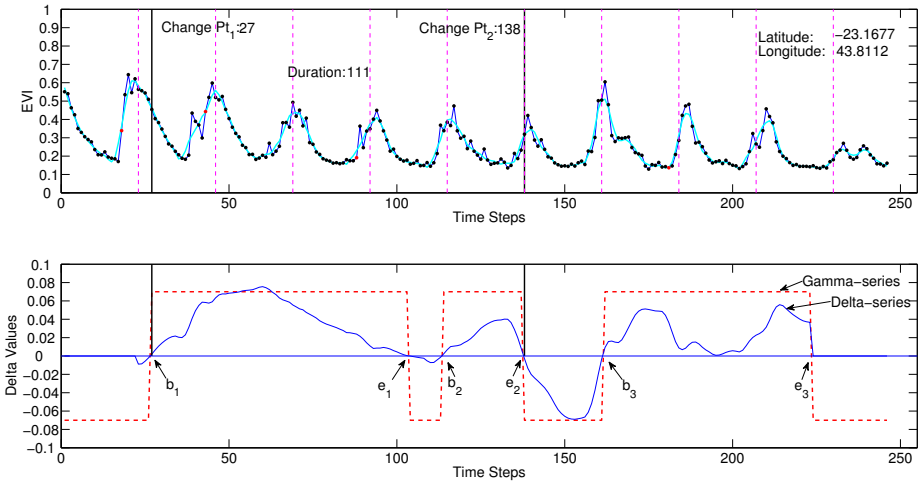


Figure 3.2: The top plot shows the original time series (line connecting the dots), smoothened time series, and the identified changed period (between solid vertical lines). The bottom plot shows the corresponding Δ-series (continuous curved line) as well as the Γ-series (broken line) scaled by a factor of 0.07.

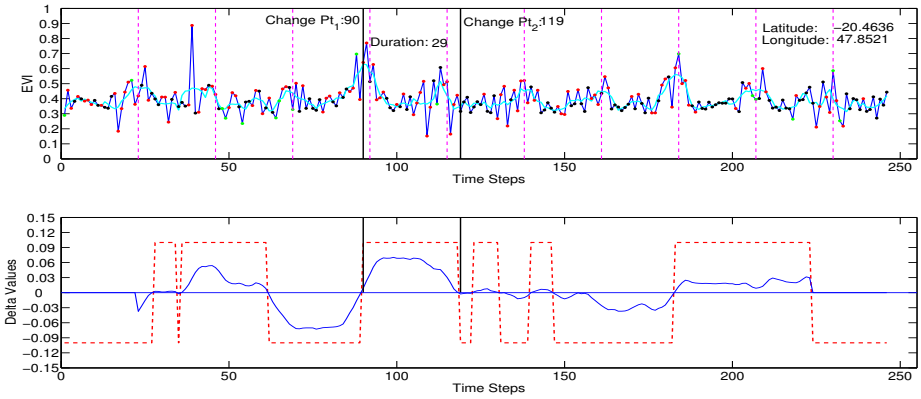


Figure 3.3: A time series in Madagascar showing a spurious rise in EVI.

.

Chapter 4

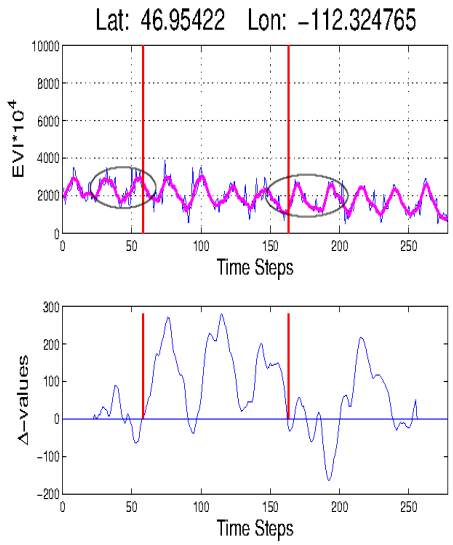
Scoring Mechanisms

In the previous chapter, we described a method to detect decreasing segments in a time series. Since any real-world time series is never completely stable (i.e. zero annual mean differences throughout the time series or in other words a zero Δ-series), every time series, including the unchanged time series, would identify multiple decreasing segments. Our objective is to identify locations on ground that have undergone a *significant* or tangible vegetation change. We suggest the following methods to tag significance to a decreasing segment.

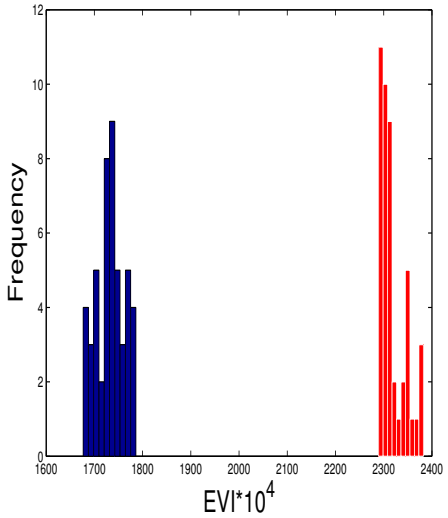
4.1 Significance Scores

Change in vegetation distribution: For a genuine decrease in vegetation, the time series pattern of the vegetation before and after the change must be different. To quantify this difference, a t-test is performed between the distribution of the time series segments before and after the decrease. Two types of distributions can be constructed:

- *Vegetation Level:* In this type, the two distributions that are constructed are of the vegetation levels before and after the change. The mean annual vegetation for few years before the change forms the samples for the first distribution while the mean annual vegetation for few years after the change ended forms the samples for the second distribution. If these distributions are determined to be different using the t-test, the vegetation is determined to be changed and the significance



(a) Top Plot: Sample EVI time series (blue), smoothened time series on which the algorithm was run (magenta), start and end change points (red). Bottom Plot: Corresponding Δ series (blue).

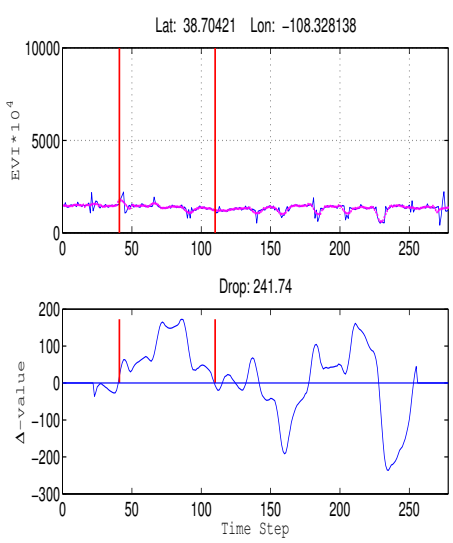


(b) Histograms of the average yearly vegetation values before the start change point (red) and after the end change point (blue).

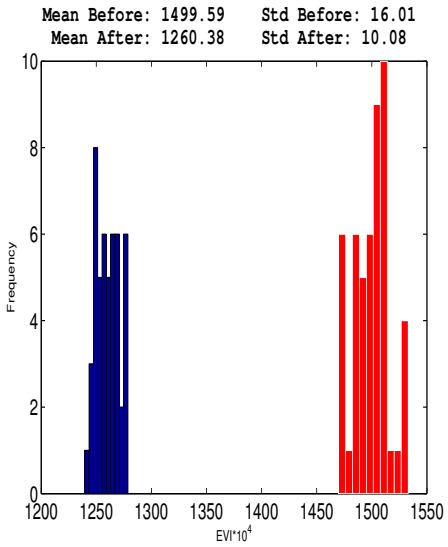
Figure 4.1: A sample time series with detected change points and average vegetation level distributions before and after the change.

of the change is denoted by the t-statistic.

- *Vegetation Difference:* In this type, the two distributions that are constructed are of the vegetation differences during the purported homogeneous and heterogeneous regions of the time series. The homogeneous region consists of the years before the change started and the time series is expected to form similar seasonal cycles during these years. The homogeneous region would not consist of the years after the change ended because there is uncertainty about the events that follow after the change: there could be a natural regrowth of vegetation, or planting of agricultural crops, or little change due to urbanization. The heterogeneous regions consist of two sets, the one that precedes the change and the one that follows the change. For a true change, the distribution constructed out of the difference



(a) Uninteresting change - Top Plot: Sample EVI time series (blue), Smoothened time series on which the algorithm was run (magenta), start and end change points (red). Bottom Plot: Corresponding Δ series (blue) with start and end change points marked (red).



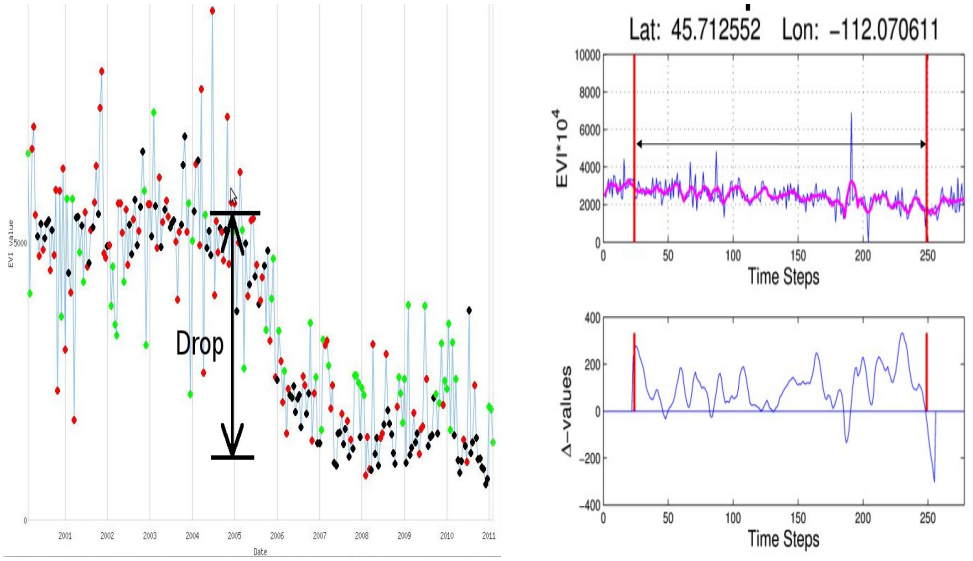
(b) Uninteresting change - Histograms of the average yearly vegetation values before the start change point (red) and after the end change point (blue).

Figure 4.2: An uninteresting time series getting high score using t-statistic.

in vegetation during the homogeneous years are expected to mean around zero, whereas the distribution constructed out of the difference in vegetation during the heterogeneous years are expected to have a much larger mean. If these distributions are determined to be different using the t-test, the vegetation is considered as changed.

Figure 4.1a shows a sample time series with detected change points and its corresponding Δ-series. Figure 4.1b shows the histograms of *vegetation levels* before the start change point and after the end change point and clearly shows that the vegetation distribution has changed. However, in some cases the drop in vegetation level can be insignificant and may not be of interest to the user even if the distribution before and after the identified change points are different. Figure

4.2a shows one such time series with Figure 4.2b showing its corresponding vegetation level distributions before and after the change. Therefore alternative scoring methods are suggested in order to better interpret and characterize a change.



(a) Using drop as the scoring method.

(b) Using duration of decrease as the scoring method. Top Plot: Sample EVI time series with start and end change points. Bottom Plot: Corresponding Δ series.

Figure 4.3: Other scoring mechanisms.

Drop in EVI: It is the difference between the mean annual EVI just before the start of the drop and the mean annual EVI just after the termination of the drop (Figure 4.3a).

Length of the Drop Window: The length can be a powerful indicator of the confidence of the change. A decrease of longer duration, even with a small *Drop in EVI*, can be of high confidence. Beetle infestation in Montana (Figure 4.3b) is a good example of this scenario.

Total Loss in EVI: If we assume that the drop didn't take place in an actually changed time series, we can suppose that the EVI pattern, P_1, representing the year before

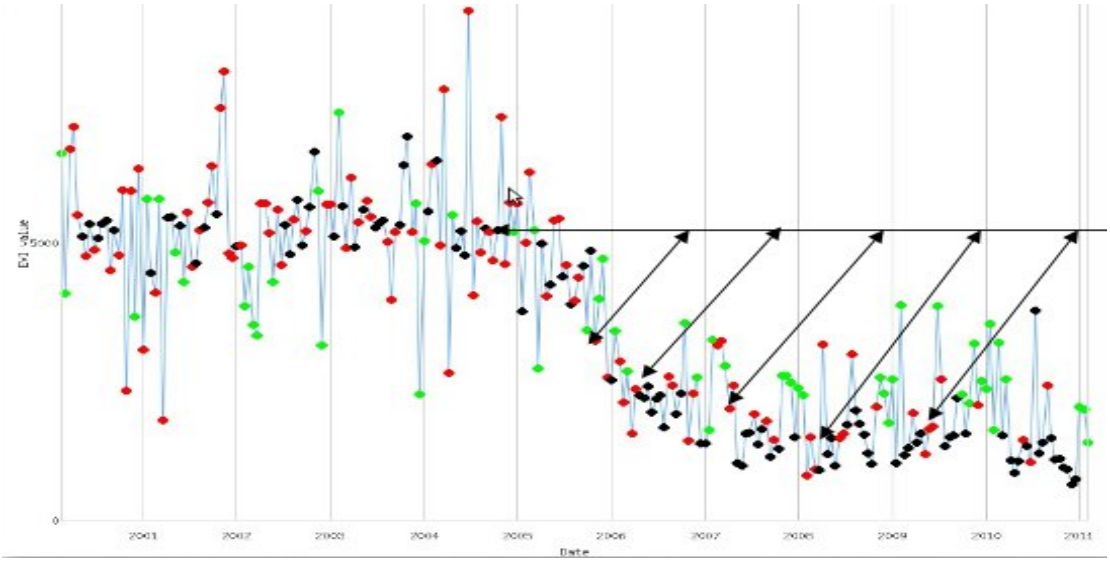


Figure 4.4: Using total loss in EVI as the scoring method.

the start of the drop would have continued. In such a scenario, the total loss in EVI would be the area enclosed between two time series, one that should have been had no vegetation loss occurred, and the other which is the actual time series in which there is a loss in vegetation (Figure 4.4).

Time taken for significant recovery: Here, we also introduce the concept of a *third* change point (the first being the drop start time step and the second being the drop end time step). After the drop occurs, if one wants to determine how long it takes for the time series to have a *significant* recovery, the third change point can be used. The third change point is positioned at a time step after the second change point such that the EVI values have risen a significant percent (say, 50%) of what it has dropped during the drop window. The position of the third change point can be an indicator of the confidence of the change because if it is realized after many time steps following the second change point, the drop is trustworthy as the vegetation stays low for a long time. On the other hand, if the third change point occurs soon after the second change point, it might mean that either the vegetation indeed recovered very quickly, or the drop was actually spurious and short-lived.

4.2 Accounting for Variability

Though we have briefly mentioned variability in a time series before, here we discuss it in the context of quantifying the EVI loss. Natural variability occurs in EVI values from one year to the other due to changes in environmental conditions such as temperature, precipitation, cloud cover, etc., or imprecision in measurement. Such a change in a time series should not be regarded as a loss in vegetation. Furthermore, a true loss in vegetation would be over and above the natural variability because a loss in vegetation equal to the natural variability would actually be a common signature of that vegetation. A way to model the natural variability is to take a mean of pairwise distance between EVI values of annual segments either during the first few years (before the first change point) or immediate previous few years before the first change point. The *city-block* distance measure (L_1 norm) works well for computing this variability. This variability is subtracted from the *Drop in EVI* in order to reflect a true drop in vegetation on the ground. Similarly, the EVI values of the pattern P_1 (refer to *Total Loss in EVI*) should be lowered by this variability before computing the *Total Loss in EVI*.

Chapter 5

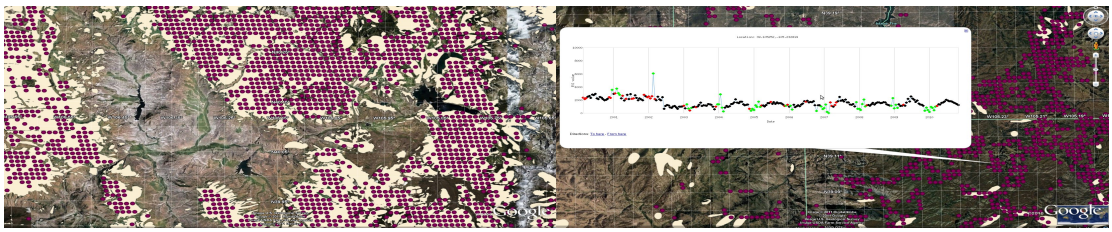
Experimental Evaluations

In this chapter, we evaluate the performance of PDELTA using real and synthetic datasets with *Total Loss in EVI* as the scoring mechanism. The real data evaluation is done in two geographically diverse regions to show the capability of this approach to detect changes in vegetation with dissimilar time series signature. On the other hand, the synthetic dataset evaluations focuses on comparing PDELTA with two existing time series change detection techniques, BFAST and CUSUM.

5.1 Real Data Evaluation

Evaluation of a scheme for detecting changes in forest cover is challenging due to the lack of high-quality ground truth. The most reliable methods for generating ground truth (e.g. ground surveys) are very expensive and are thus only available for small regions.

In the absence of such gold standard ground truth, less reliable labels generated by some other scheme or via aerial surveys can still be used for validation, but care must be taken to check if "false positives" (i.e. changes found by the scheme but not in the validation data) are indeed false, since they could have been missed by the scheme used to generate labels. Similarly, one needs to check if "false negatives" (i.e. changes noted in the validation data but not found by the scheme) are indeed changes on the ground as they could be incorrectly classified as changed by the other scheme.



(a) A snapshot showing locations identified as changed by the proposed approach (red circles) overlaid on the validation data polygons.

(b) Snapshot showing a large region in Colorado where there was a drop in EVI due to Fire in the year 2002. Most of this region is not covered by the polygons.

Figure 5.1: Snapshots showing regions in Colorado where vegetation loss was detected.

We evaluated our approach on two regions; Northern Colorado and Southern Madagascar for which moderate quality labels are available in the form of polygons covering degraded areas of the forests. These regions are interesting because they have completely different vegetation types, and the degradation is caused by different mechanisms; specifically, insect damage in Colorado, and logging in Madagascar. During analysis in both regions we emphasize the strength of this algorithm in detecting changes that are difficult to identify, as well as the capability of this approach to capture many changes that are missed by the validation data.

5.1.1 Evaluation on Colorado Forests

The first region of analysis is forests in northern Colorado (the region bounded by 39°N—41°N; 108°W—104.5°W). The US Forest Service and its partners [15] maintain data sets which map the regions of forest cover that have degraded between years 2002 and 2008 in northern Colorado. The objective was to detect regions of forest degradation using our approach and evaluate it against the above validation data (which has been transformed to polygons). We present our analysis as follows:

True positives are points detected by our approach that also lie in the polygons. As seen in Figure 5.1a, there is a very good overlap of the detected points with the validation polygons. The algorithm is also able to correctly identify the period of degradation. Colorado is a difficult region because changes here are often very gradual, sometimes

to the extent that there is no visible change in EVI signal upon manual inspection. Nevertheless, our approach identified a significant number of points. Figure 5.2 shows the typical EVI time series in this region.

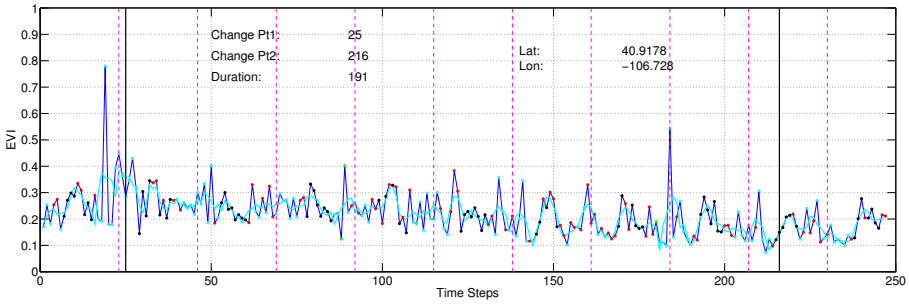


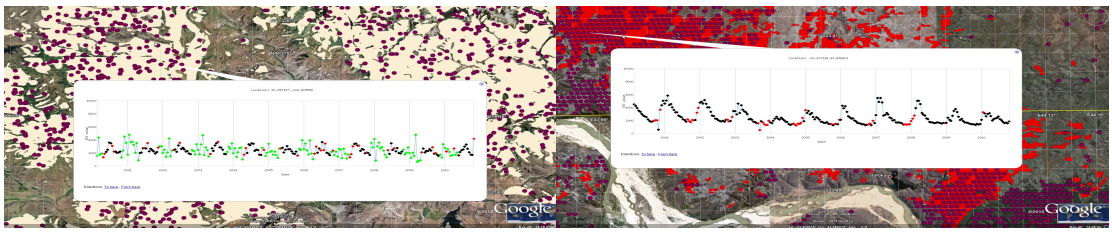
Figure 5.2: A typical gradual drop in Colorado due to beetle infestation.

False positives are points that we detected as change but do not lie in any of the polygons. There could be several reasons for this: (i) the decrease in vegetation in these areas is caused by factors other than those considered in constructing the validation data; (ii) it is known that the polygons can be inaccurate (iii) these points are in fact not changed, but due to noise in EVI appear as changed and thus given a high score by the proposed approach. Our manual inspection shows that majority of false positives with high scores are due to (i) and (ii). For example, consider the region shown in Figure 5.1b. This region is not part of any polygon even though the change is quite apparent and is likely due to fire [16].

False Negatives are points that we did not detect as changed but which lie inside the polygons. Figure 5.3a, shows an example of such a time series. This time series does not show any change in the EVI signal. There are numerous time series in this region that show little perceptible change and are in the polygons. So either the vegetation loss here is too gradual to be detected by our approach, the change on the ground is not captured by the EVI signal, or the polygons are inaccurate.

5.1.2 Evaluation on Madagascar Forests

The second region of analysis is southern Madagascar (the region bounded by 25.6°S—20°S). The validation data is obtained from Center for Applied Biodiversity Science



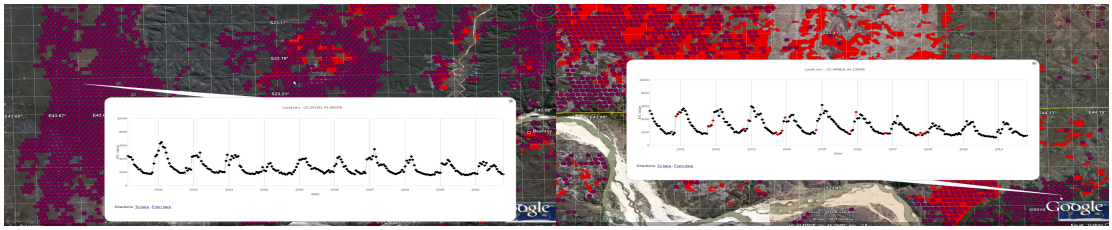
(a) This figure shows an example time series in Colorado that has no perceptible change but which fall inside the ground truth polygon.

(b) Time series of a region in Madagascar showing gradual decrease in EVI starting from year 2001, which lie inside the ground truth polygon.

Figure 5.3: Snapshots of false negatives in Colorado and true positives in Madagascar.

(CABS) at Conservation International (CI), whose analysis is based on bitemporal Landsat image comparison between years 2001 and 2005 [17]. Hence, the validation data (or polygons) cover changes only between these two years.

True Positives and False Negatives: Figure 5.3b shows an example of a true positive. The image clearly shows the gradual drop starting in the year 2001. Most points in the validation polygons show similar behavior, although with varying decay rate and duration. Hence, effectively there are few false negatives since most points in the validation polygons can be found by our algorithm.



(a) A snapshot showing a large region in Madagascar where vegetation loss started to occur in the year 2001.

(b) This figure shows an area where vegetation loss occurred after year 2005.

Figure 5.4: False Positives in Madagascar having a decreasing EVI signal.

False Positives: Most of the false positives were observed due to the following reasons:

(1) Vegetation degradation occurred during the period of analysis (2000-2005) but the vegetation recovered during 2005, which causes it to be missed by the technique used

in generating the validation data (Figure 5.4a). The entire cluster of points to the left in the figure has similar time series signature but lies outside the validation polygons. This illustrates the limitation of the techniques that are based on the comparison of images taken on two different dates. (2) Significant vegetation degradation is visible only after 2005 hence was not included in the validation data set. Figure 5.4b shows an example of such a region. This type of identification also highlights the capability of our approach to find changes with high temporal precision in a continuous manner. This is opposed to the image-based methods where analysis is usually done on snapshots of images generally few years apart.

5.2 Simulated Data Evaluation

In this section, we compare our proposed technique with two other approaches for gradual change detection, CUSUM and BFAST [8] using data sets with simulated noise and change characteristics. The BFAST technique is designed to detect long-term changes in satellite image time series. It decomposes a time series into trend, seasonal, and residual components, such that the intra-segment models are constant, while inter-segment models are dissimilar. BFAST identifies the optimal position of trend and seasonal breakpoints by minimizing the residual sum of squares (RSS), and the optimal number of breaks can be determined by minimizing an information criterion. Before estimating the breakpoints, the ordinary least squares residuals-based moving sum test is used to identify if any breakpoints are occurring in the time series. As output, BFAST provides the trend breakpoints and associated trends, seasonal breakpoints and associated seasonal models, and logical values indicating whether the time series is considered changed in the seasonal or trend components. We do not consider the seasonal component in this study since we are looking for a decreasing trend.

Evaluating our algorithm against BFAST is not straightforward since BFAST looks not only for drops, but any type of trend change in a time series. Also, simply consulting the logical vector values that labels a time series as changed was not feasible for two reasons: (i) BFAST appears to be sensitive to noise and frequently finds different trends even in a stable time series and labels them as changed. (ii) BFAST would label a time series as changed if any type of trend change is present, notwithstanding the absence

of a decreasing trend. In addition, BFAST also requires some parameter settings such as the minimum segment size and maximum number of breakpoints desired. These parameters are not mandatory, but not setting them makes it quite sensitive to noise, resulting in breaking even a single trend into multiple segments. Therefore, construction of the synthetic datasets had to be in consonance with the parameter values of BFAST.

We constructed two types of datasets, DS1 and DS2, the first containing three different trends (two trend breakpoints), and the other containing four different trends (three trend breakpoints). The maximum number of breakpoints set in BFAST for these two types of datasets were two and three respectively, and it was expected that BFAST would correctly identify all the given trends. Since we are interested in identifying the decreasing trend, our trend of interest among the ones returned by BFAST is the one which has the largest decrease across it. Its change score (which also represents the score of the time series) is computed in the same manner in which we compute the score for our proposed approach. Below, we describe the creation process of the synthetic dataset.

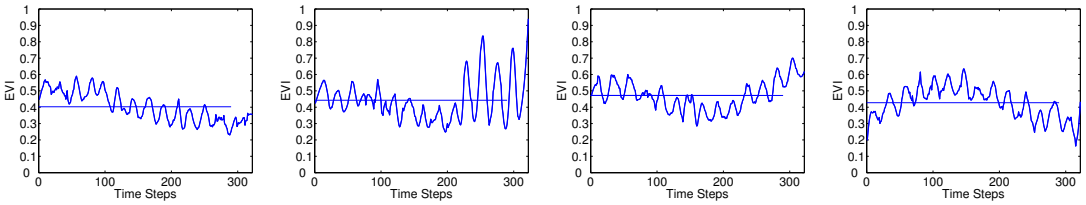
5.2.1 Synthetic Data Generation

The datasets DS1 and DS2 are comprised of 1100 time series each, in which a gradual decrease phase was inserted in 80 time series for DS1, and 120 time series for DS2. In DS1, 40 time series also have an increasing trend. The remaining stable time series in both the datasets are identical (total: 980). Each time series has 322 time steps, with a seasonal period of 23 time steps (in order to mimic the MODIS EVI time series having 14 years of data). The seasonality in a time series is created using a function of the form:

$$F(x) = A * e^{\frac{-|x-m|}{B}}$$

where A controls the amplitude, x varies between time steps of a particular year, m controls the position of the peak in that year, and B controls the curve. The shape of $F(x)$ mimics a typical seasonal vegetation pattern of a forested region (or farming cycle) as reflected in an EVI time series.

Each time series has different types of noise added to it. We define these below, followed by the characteristics of the changed and stable time series.



(a) Pattern one (b) Pattern two (c) Pattern three (d) Pattern four

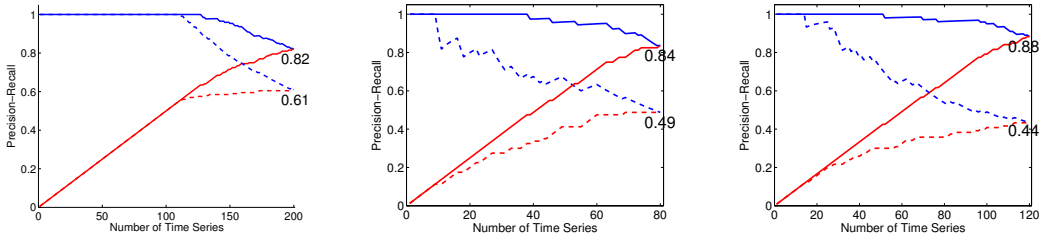
Figure 5.5: Different decreasing patterns in D0 dataset. Each pattern has fifty samples (total two-hundred). The horizontal line shows the mean of the time series.

Noise characteristics Two types of noise are introduced in the dataset. w_1 is white noise that is added to each time step in the time series. w_2 is outliers, that results in very high (upward spikes) or very low (downward spikes) values at certain time steps as compared to that of its neighbors.

Characteristics of a changed time series There are three phases in these time series. (1) *beforePhase* is the period in the time series before a drop. Here, seasonal cycles (pattern during one year) are represented by $F(x)$. This phase may have an increasing trend or a stable trend. Noise w_1 and w_2 is added to each time step. All introductions in this phase are probabilistic as a Gaussian distribution within sufficient ranges specified in advance. This includes the values of w_1, w_2, duration of this phase, height of the data values during each year, duration of the increasing trend if any. (2) *changePhase* starts as soon as *beforePhase* ends. The majority of these time series have a decreasing trend. The base level of successive years in this phase is reduced gradually from starting of this phase till its end. The duration of this phase and the amount of drop introduced are probabilistic within a certain range. Noise w_1 and w_2 are added to this phase as well. In a small fraction of time series, an increasing trend is added during this phase instead of a decreasing trend to include more variety of time series. However, since these time series do not contain a decreasing trend, they are considered as false positives if detected by any algorithm. (3) *afterPhase* starts after the *changePhase* ends. Each year in this phase is also represented by $F(x)$ with w_1 and w_2 added.

Characteristics of a stable (unchanged) time series These time series have one phase with a constant level whose value is probabilistic within a certain range.

Each seasonal cycle in these time series is also represented by $F(x)$ with noise w_1 and w_2 added.



(a) DS0 - Precision-Recall for PDELTA (solid - precision: 0.82) and CUSUM (broken - precision: 0.61).

(b) DS1 - PDELTA (0.84). BFAST (0.49)

(c) DS2 - PDELTA (0.88). BFAST (0.44)

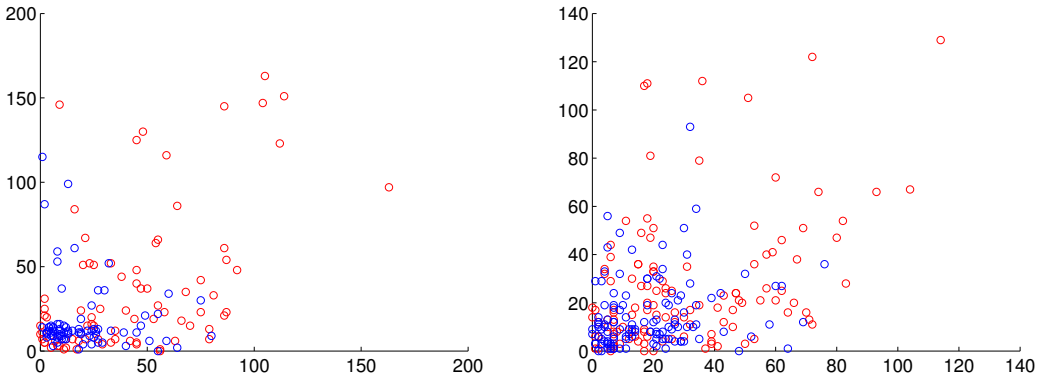
Figure 5.6: Precision curves (blue) and recall curves (red) for PDELTA (solid curves), CUSUM and BFAST (dashed curves). Noise $w_1 = 200$, $w_2 = 10\%$.

5.2.2 Evaluation Strategy

PDELTA, CUSUM, and BFAST are applied to these datasets after preprocessing as described in Section 2. We compare the performance of our approach with CUSUM and BFAST separately. It is because these two approaches return different information about the drop, and we adjust our evaluation according to this information. For CUSUM, we combine the samples of time series from datasets DS1 and DS2 into a single dataset DS0.

5.2.3 Comparison with CUSUM

We evaluated the performance of PDELTA and CUSUM on the dataset DS0. There are four different types of time series in DS0 that have a decreasing trend (Figure 5.5). Each pattern has 50 different samples. Overall, this dataset has 200 decreasing time series, and 1020 stable time series (Total: 1220). The time series patterns included in DS0 are common in the real world datasets. Pattern one (Figure 5.5a) is an example of a stable forest that degrades over many years. Pattern two (Figure 5.5b) is an example of



(a) DS1 - PDELTA (Blue), BFAST (Red).

(b) DS2 - PDELTA (Blue), BFAST (Red).

Figure 5.7: Scatter plots of the deviation of change points detected by PDELTA and BFAST from actual drop start (x-axis) and drop end (y-axis) time steps.

conversion of forests to farm lands, as could be noticed from the typical farming cycles during the later part of the time series. Figure 5.5c reflects some plantation following a deforestation. Figure 5.5d could depict a failed reforestation.

The precision-recall curve of the result of the two algorithms is shown in Figure 5.6a. CUSUM performed best on the samples of the pattern shown in Figure 5.5a. Admittedly, CUSUM performs well for any gradually decreasing time series that accumulates a large score during the beginning time steps, which would happen if most of the beginning values are placed above the expected value μ. However, CUSUM would perform equally poorly on time series having an opposite signature. This highlights a major drawback of CUSUM. Consider the patterns two and three shown in Figures 5.5b and 5.5c, on which CUSUM performed poorly. In these time series, most of the values in the beginning are below μ, which is taken as the mean of the time series, implying that the majority of the later values are above μ. Such time series would never be able to accumulate a high cumulative sum since it incurs a large loss in the beginning due to negative deviations, and therefore would be given a low score. Many such scenarios could be constructed where there is a decreasing trend present but the time series never accumulates a high enough score for it to be significant.

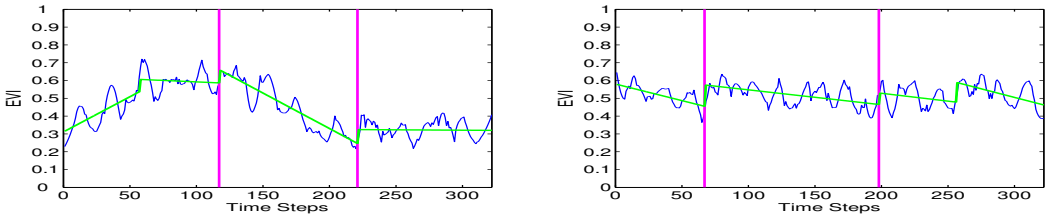
We also investigated alternative ways of computing the expected value, but these

variations either repeated some of the above drawbacks, or other drawbacks were discovered in them. For instance, by taking μ as the mean value of the first year, CUSUM performed poorly on patterns two, three, and four (Figure 5.5). Note that if we consider this variation, we are looking for the minimum cumulative sum (instead of the maximum) since an ideally decreasing time series would have a highly negative cumulative sum. The main disadvantage of this variation is that the cumulative sum is highly dependent on the first year values. If the first year values are noisy, it can drastically affect the algorithmic output. As an example, if μ is even slightly high due to noise, a stable time series could get a high score. In contrast, if μ is low, decreasing time series can go undetected (Figure 5.5d).

The precision-recall curve suggests that PDELTA performed considerably better than CUSUM on this dataset. Additionally, PDELTA also identifies the period of decrease.

5.2.4 Comparison with BFAST

Our evaluation with BFAST is based on two factors: (1) Precision-recall curves for PDELTA and BFAST, formed by ranking the time series in decreasing order of scores, when evaluated on DS1 and DS2 (Figures 5.6b and 5.6c). (2) Scatter plots of the deviation of the change points identified by the two approaches from the actual positions where the breakpoints were introduced in the synthetic datasets (Figure 5.7).



(a) Correctly detecting a decreasing trend. (b) Many trends identified in stable time series.

Figure 5.8: Trends using BFAST. Vertical lines identify the period of maximum drop.

The scatter plots show that the distribution of the deviation of the change points identified by PDELTA is closer to zero than that of BFAST. BFAST segments a time series based on RSS and a Bayesian Information Criteria (BIC), which has no bias towards

identifying decreasing periods. If identifying the decreasing trend as a separate segment minimizes RSS, BFAST will correctly identify the decreasing trend. Otherwise, it may combine a part of the decreasing trend with an adjacent trend. Also, BFAST appears to be sensitive to noise in a time series. It correctly detected the trends introduced for many time series (Figure 5.8a), but it segmented stable time series into different trends as well (Figure 5.8b). This suggests that BFAST might not be very suitable for highly variable time series, where noise levels can distort the ideal seasonal pattern enough. Such highly variable time series are characteristic of the tropical belt such as in Para (Brazil), Peru, Congo, etc.

Chapter 6

Qualitative Case Studies

In the previous chapter, we discussed the application of our approach to detect Madagascar logging and Colorado insect infested forests as well as performed a comparative evaluation against two existing time series change detection methods. Being scalable, we applied PDELTA to detect vegetation changes across the globe. In this chapter, we discuss a selected sample of other regions where we found interesting changes. In addition, we do a comparison of PDELTA with CUSUM on a real EVI dataset.

6.1 British Columbia, Montana, Wyoming

Mountain Pine Beetle Infestation

In addition to Beetle infestation in Colorado, we detected thousands of square kilometers of trees infested by beetle in British Columbia, Montana and Wyoming. Trees infested by mountain pine beetle develop white, pink or brown "pitch tubes" on the trunk followed by the foliage turning yellowish to reddish throughout the tree crown in about eight to ten months after a successful mountain pine beetle attack (`http://beetles.mt.gov/Biology/Damage.asp`). Figures 6.2a, 6.2b, 6.2c, 6.2d show images of beetle infested areas and beetles at work. These images were obtained from Ministry of Forests, Lands and Natural Resource Operations at British Columbia (`http://www.for.gov.bc.ca/hfp/mountain_pine_beetle/bbphotos.htm`).

The ministry of forests in British Columbia states that 710 million cubic meters of timber has been cumulatively killed by mountain pine beetle and a cumulative area of

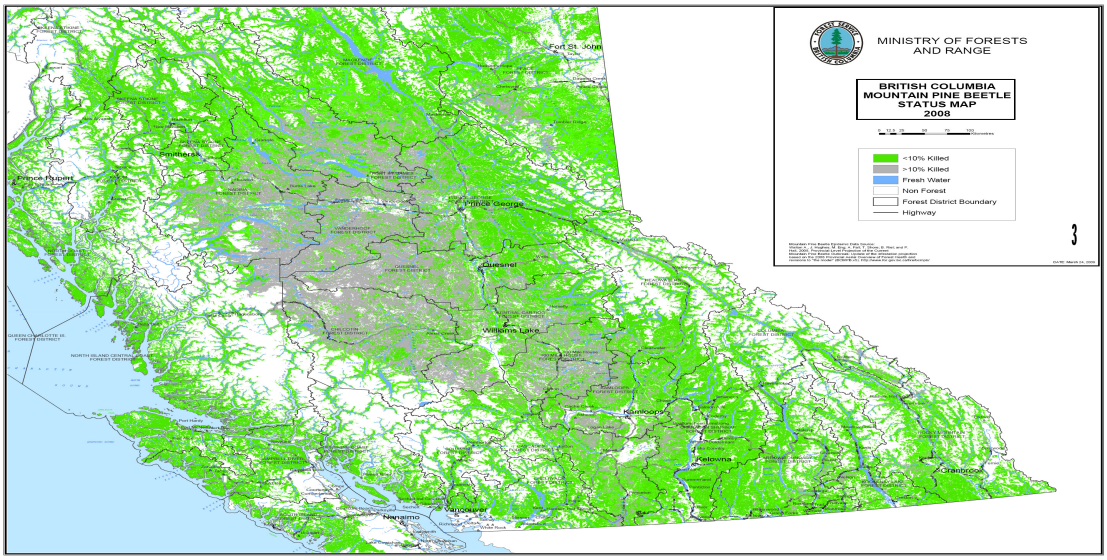


Figure 6.1: Mountain Pine Beetle 2008 status map. *src:* ***https://www.for.gov.bc.ca/hfp/mountain_pine_beetle/maps/Culm_kill_2008.jpg***

about 18.1 million hectares (approximately 180,000 km^2) has been affected to some degree (`https://www.for.gov.bc.ca/hfp/mountain_pine_beetle/facts.htm`). Figure 6.1 shows a 2008 status map of mountain pine beetle affected areas. PDELTA detected over 50,000 square kilometers of vegetation-degraded areas in British Columbia. A sample of this area is shown by pixel-converted-polygons in Figure 6.3.

Figures 6.4 shows snapshots of a region in Montana detected as changed by PDELTA using the *change in vegetation distribution* scoring mechanism (chapter 4). Each pixel is of 1km spatial resolution. Thousands of square kilometers of forests were detected as degraded. Due to the unavailability of validation data, we were unable to confirm the cause of degradation. However, Montana state government website states that insect infestation is one of the main causes of vegetation disturbance in Montana and mountain pine beetle (native to Montana) has killed 800,000 acres of Lodgepole pine per season during the recent years (`http://deq.mt.gov/ClimateChange/NaturalResources/Forestry/forestryClimateChangeEra.mcpx`). In addition, brown patches of foliage are observed in most of the result set which is a good indicator of insect infestation. Furthermore, the detected time series generally have a very gradual decrease over multiple years - a characteristic signature of insect infested vegetation. Also, the close up view of the

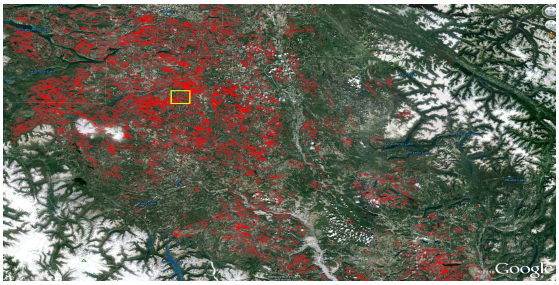
(a) Aerial view of extensive attack by mountain pine beetle.

(b) Ground view of crown of lodgepole pine killed by mountain pine beetle.

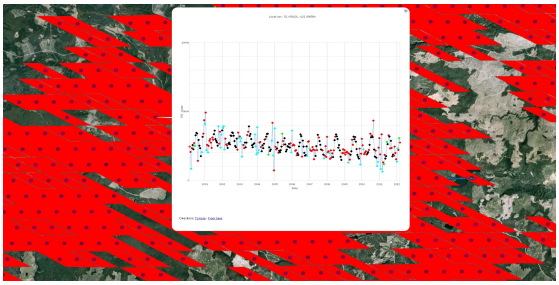
(c) Trunk of lodgepole pine showing "pitch tubes" from a mountain pine beetle attack.

(d) Adult mountain pine beetles.

Figure 6.2: Trees Destroyed by mountain pine beetle in British Columbia. *Src. of images: Ministry of Forests, Lands and Natural Resource Operations at British Columbia (`http://www.for.gov.bc.ca/hfp/mountain_pine_beetle/bbphotos.htm`).*

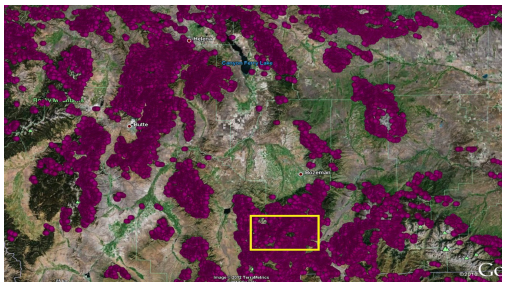


(a) Red polygons show the degraded pixels detected by PDELTA

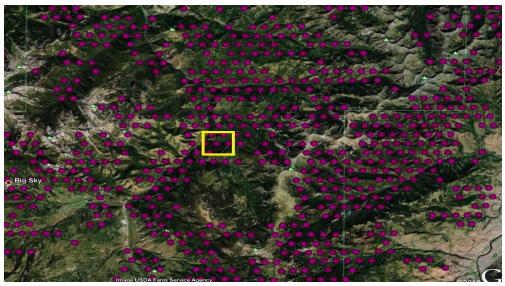


(b) Time series profile of a mountain pine beetle infested pixel.

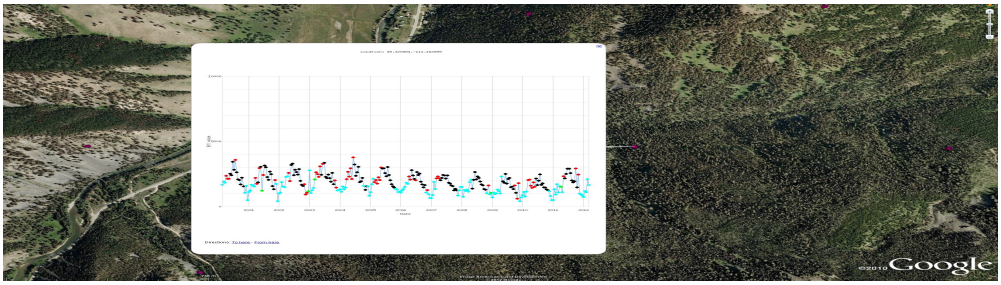
Figure 6.3: PDELTA detected thousands of square kilometers of insect infested forests in British Columbia.



(a) Google Earth snapshot of degraded forests in Montana.



(b) Close up view of an area.



(c) Time series showing vegetation decay. Brown foliage can be observed indicating insect infestation.

Figure 6.4: Regions in Montana detected as degraded.

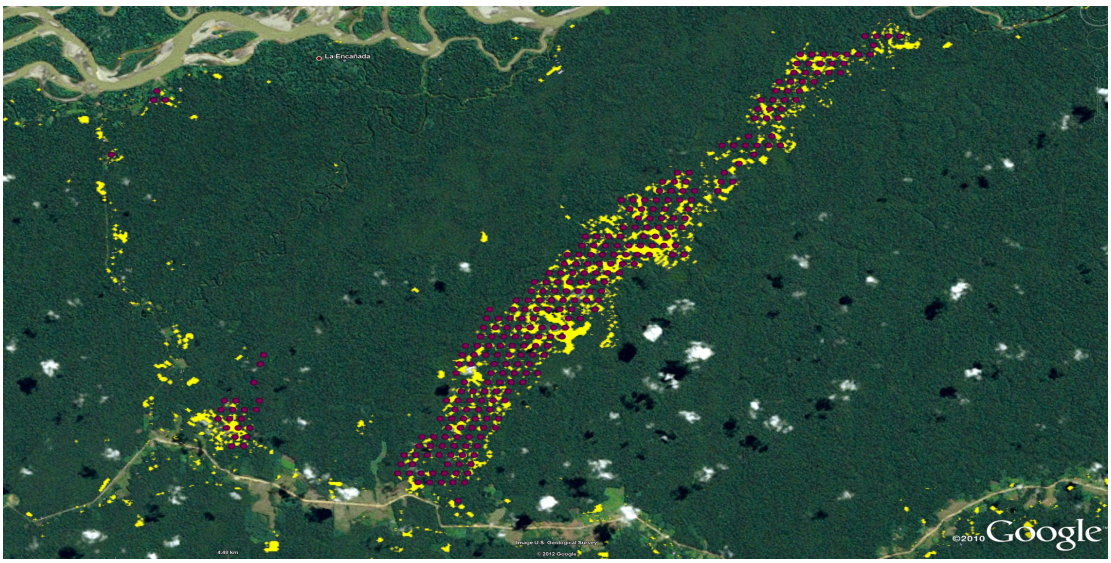
degraded regions reveal an absence of logging patches indicating the cause of vegetation decay is other than logging.

6.2 Peru

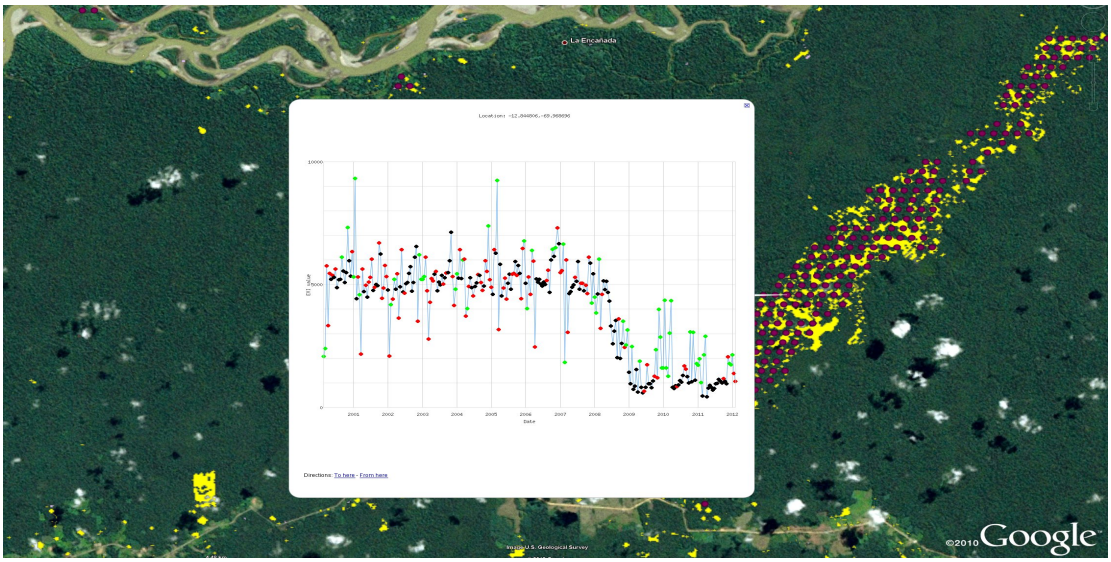
Logging

Peru has the fourth largest tropical forest area after Brazil, Congo and Indonesia (`http://news.bbc.co.uk/2/hi/7768226.stm`). Madre de Dios state in Peru has been subjected to an increased demand of land use activities due to the increase of artisanal gold miners, new timber concessions and the development of interoceanic highway [18]. [18] extensively studied deforestation in Madre De Dios state at a very high spatial resolution (0.1 hectare - 30 meters). It analyzed carbon stocks (forests) and emissions (deforestation and forest degradation) over 4.3 million hectares of land in the Peruvian Amazon. It used airborne LiDAR (Light Detection And Ranging) combined with field plots and satellite data to generate high resolution forest maps of carbon stocks and emissions. We used their deforestation and forest degradation maps as validation data to qualitatively validate our results. Because of the high spatial resolution of their product, we used the highest available MODIS EVI time series dataset of 250 meters resolution instead of our conventional 1 km resolution in order to match them more accurately. Figure 6.5a shows a sample location showing overlap between their product (yellow polygons) and EVI pixels detected by PDELTA. The corresponding time series (Figure 6.5b) shows that this region had undergone massive deforestation.

In addition, we found locations where our approach found changes while the validation data did not. Figure 6.6 shows an example. This also highlights the comprehensive nature of our approach to detect apparent changes which is not limited by coverage or scalability. On the contrary, we also found places where the validation data has polygons while we did not detect any changed pixels. These broadly fall into two categories. In the first category, the change recovers too quickly for our approach to assign high significance (6.7a). It becomes very difficult to even differentiate such changes from noise through visual inspection (Figures 6.8a, 6.8b) This rapid vegetation recovery is a common characteristic in the tropical forests which, combined with highly noisy time series signal, makes it very challenging for an approach to detect such events. The



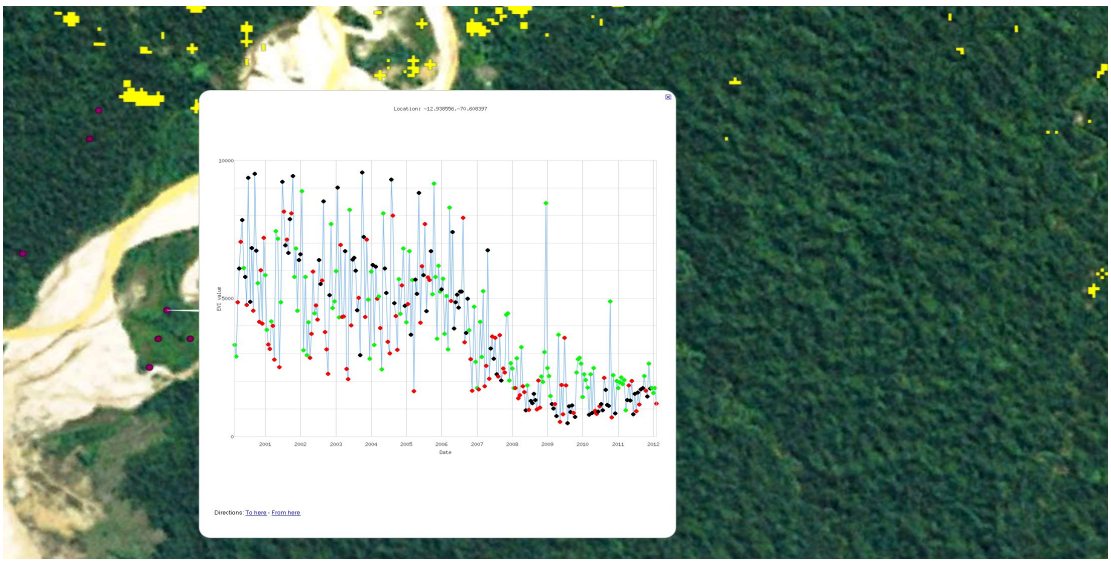
(a)



(b)

Figure 6.5: A large region where PDELTA results overlap with the validation data.

second category of changes which we did not detect were the ones where there was no apparent change in the EVI signal (6.7b). The likely possibility for this to occur is that such changes were visible at higher resolution (30 meters) of the validation data but not



(a)

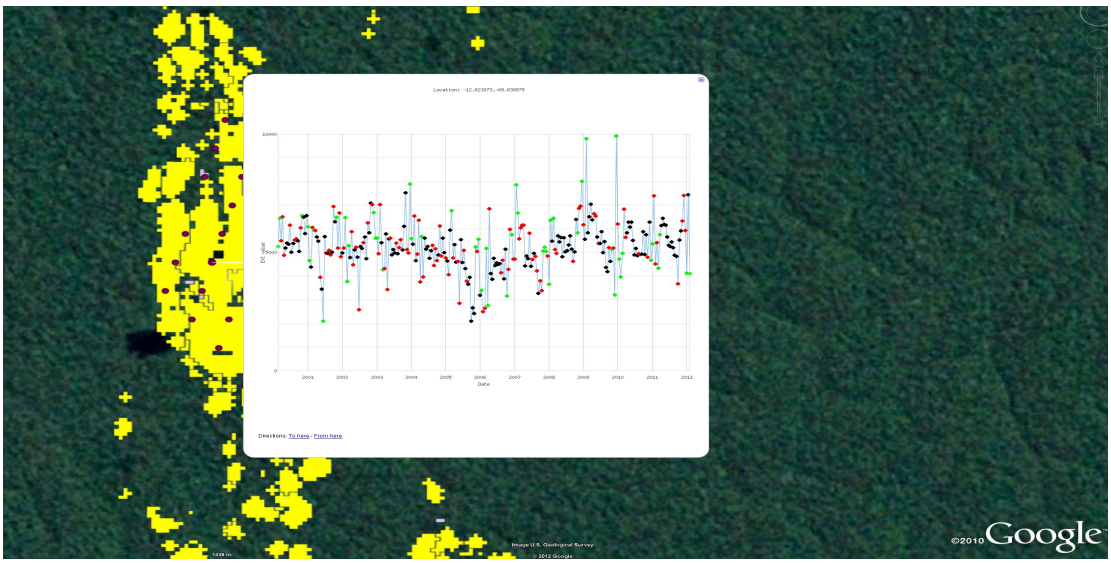
Figure 6.6: Pixel detected by PDELTA but not by the validation data.

visible at our 250 meters EVI data. The other possibility is that their approach made error in classifying such locations as degraded.

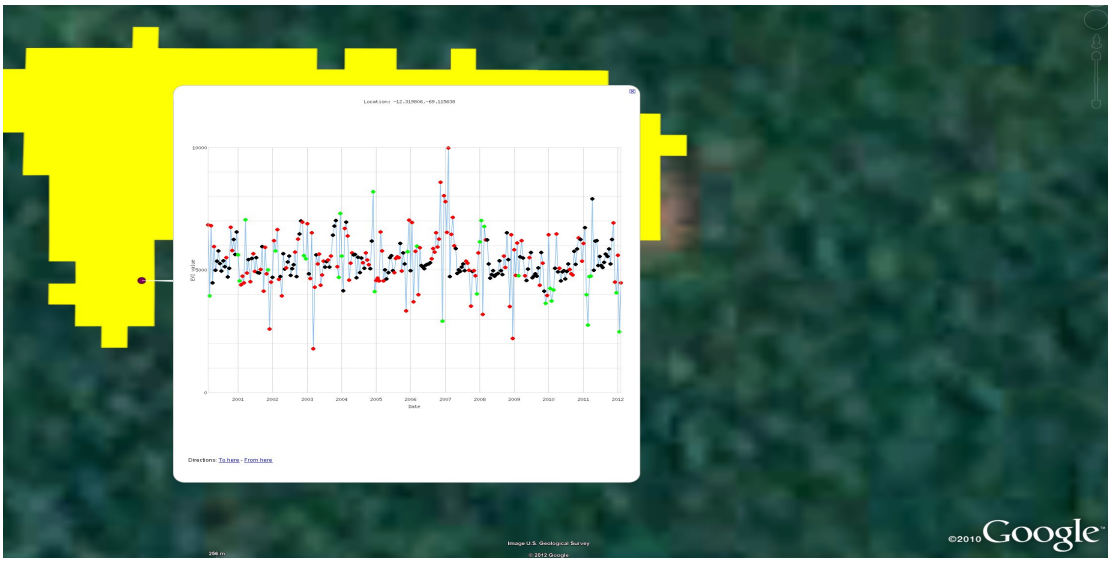
6.3 PDELTA vs CUSUM

In this section, we compare PDELTA with CUSUM in Montana and Wyoming regions and analyze their performance. Our objective is to understand the strengths and weaknesses of both the techniques on different types changes. Due to the absence of validation data in this region, we could not directly compare the results using precision-recall curves. Head to head comparison with CUSUM is in part difficult due to the selection of manual score thresholds for both the approaches which is prone to human judgement and error. Therefore we adopt an alternative qualitative comparison methodology in which we analyze the types of changes each method is biased towards as well as the changes that go undetected by one method but not the other.

.We scored the 1.44 million time series in these regions using CUSUM and PDELTA approaches. For PDELTA, we used *change in vegetation distribution* scoring method



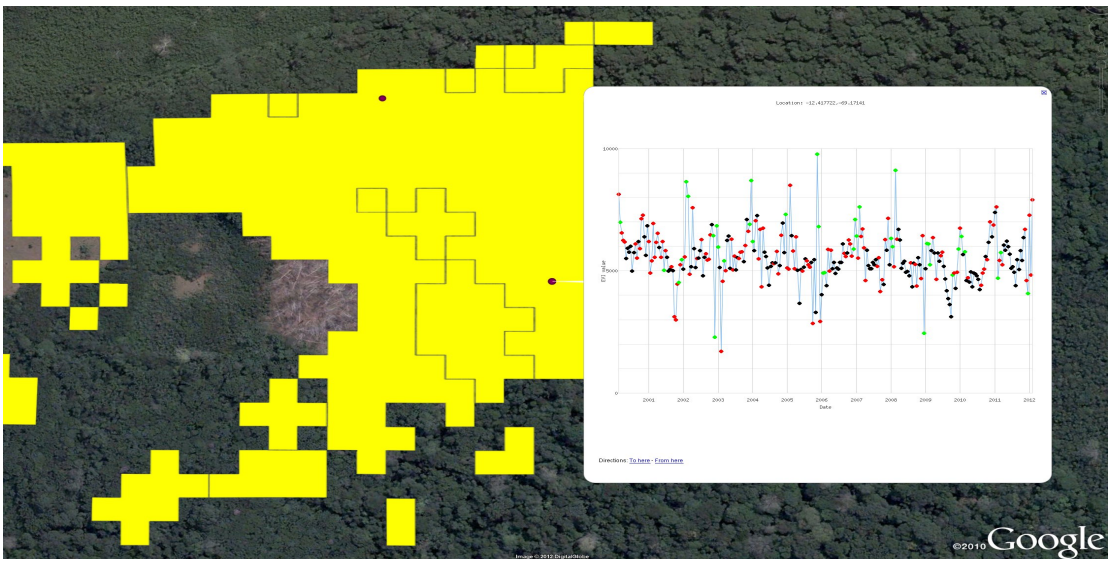
(a) Pixel not detected by PDELTA but overlap with the validation data. Cause: rapid recovery.



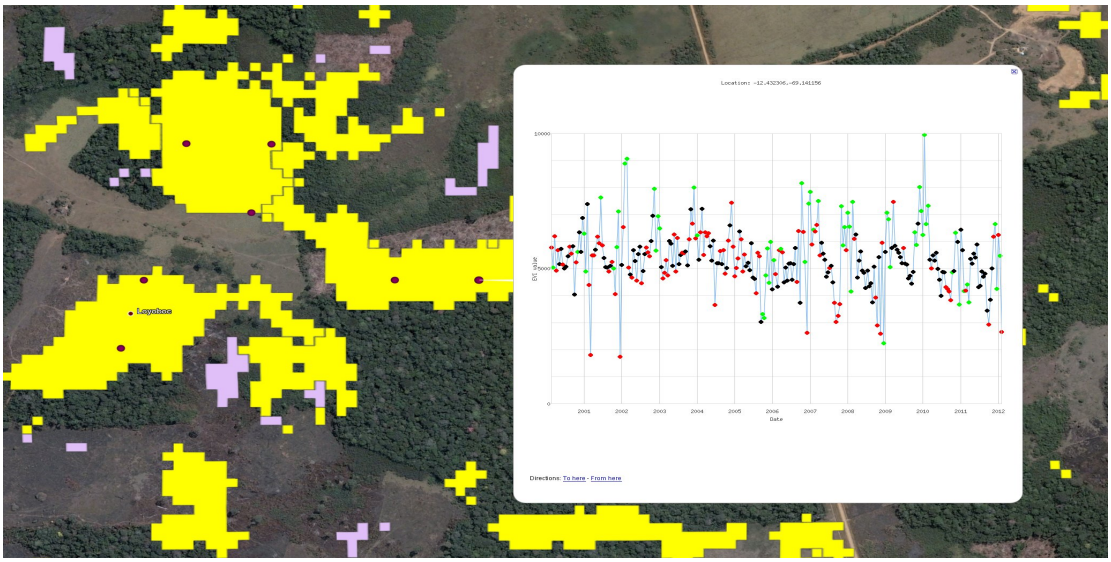
(b) Pixel not detected by PDELTA but overlap with the validation data. Cause: no apparent decrease in the EVI signal.

Figure 6.7: Pixels not detected by PDELTA but overlap with the validation data.

with t-statistic value as the score. We then ranked both scores in decreasing order and identified pixels (or time series) with high confidence of change (or top h scoring pixels)



(a)



(b)

Figure 6.8: Pixels not detected by PDELTA but overlap with the validation data. Cause: difficult time series changes to identify even through manual inspection.

from each approach. We introduced a buffer of size $\alpha * h$ on the rankings thereafter which could potentially contain changed or unchanged time series. The pixels included

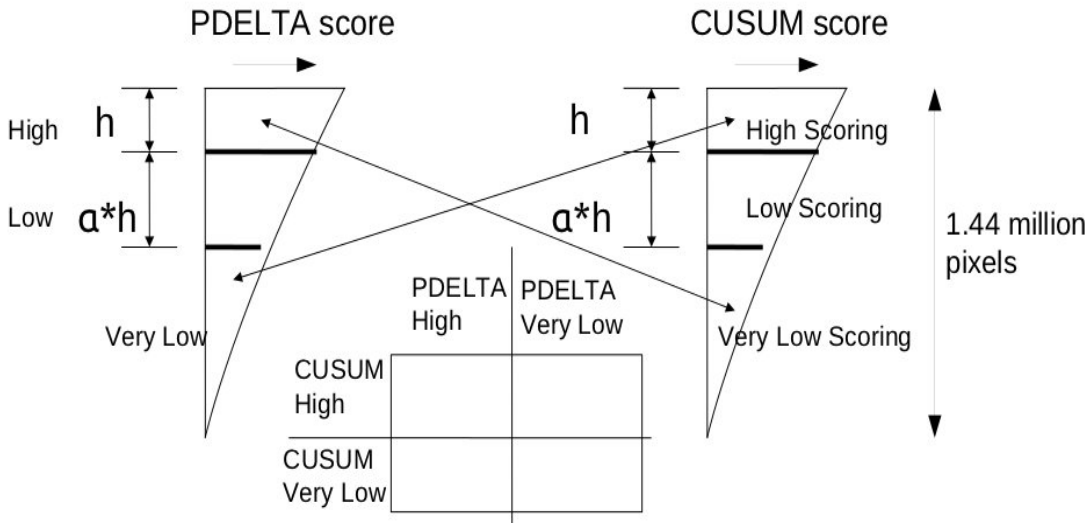


Figure 6.9: Qualitative evaluation methodology to compare two competitive approaches.

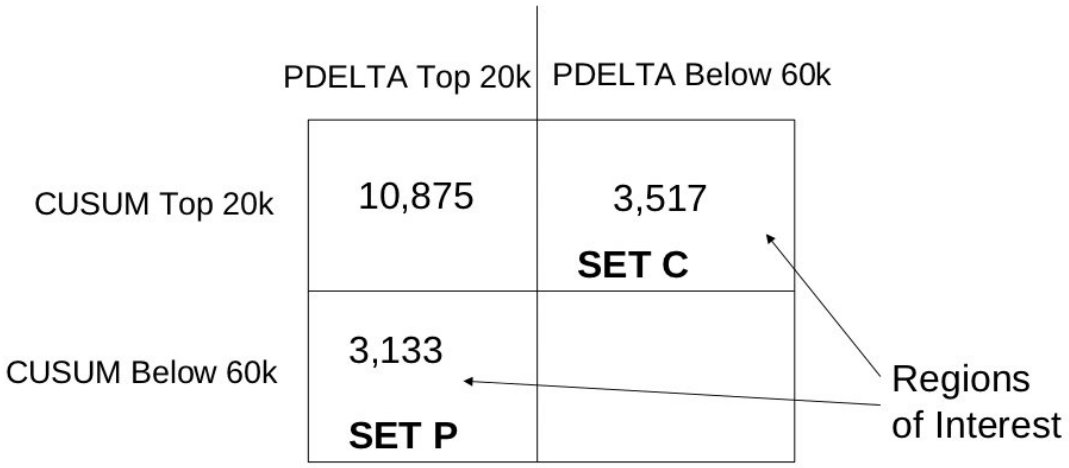


Figure 6.10: Confusion matrix of pixels with different rankings by two competitive approaches.

in this set are termed as low confidence changes. The pixels in the remaining lower ranks are termed as very low confidence or insignificant changes. We cross-checked the pixels that were given a high score by one approach and very low score by the other.

Figure 6.9 depicts this comparative evaluation methodology.

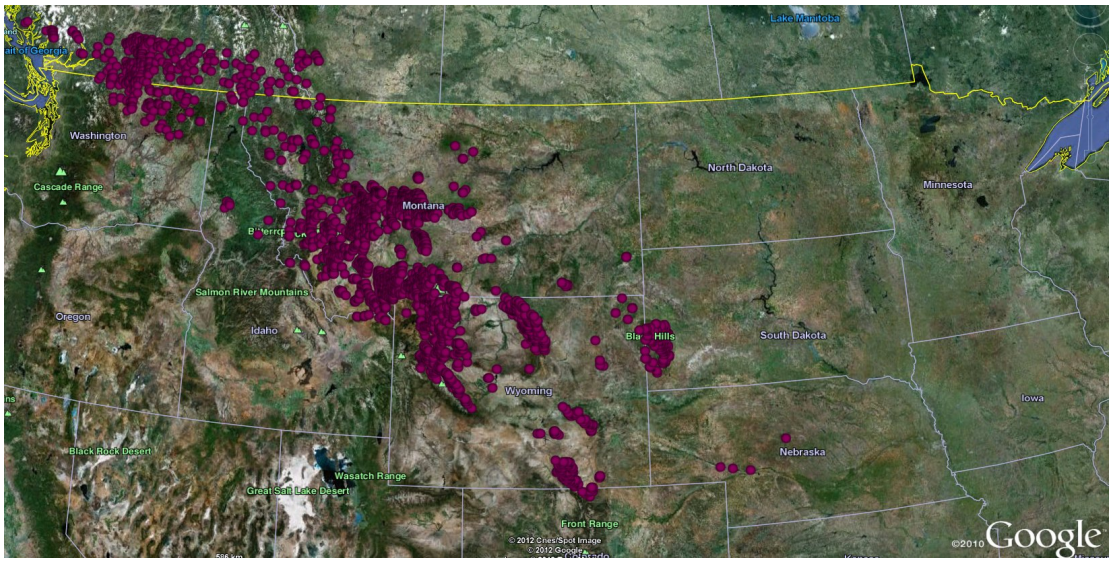


Figure 6.11: Google Earth snapshot of pixels with high scores by PDELTA but low scores by CUSUM.

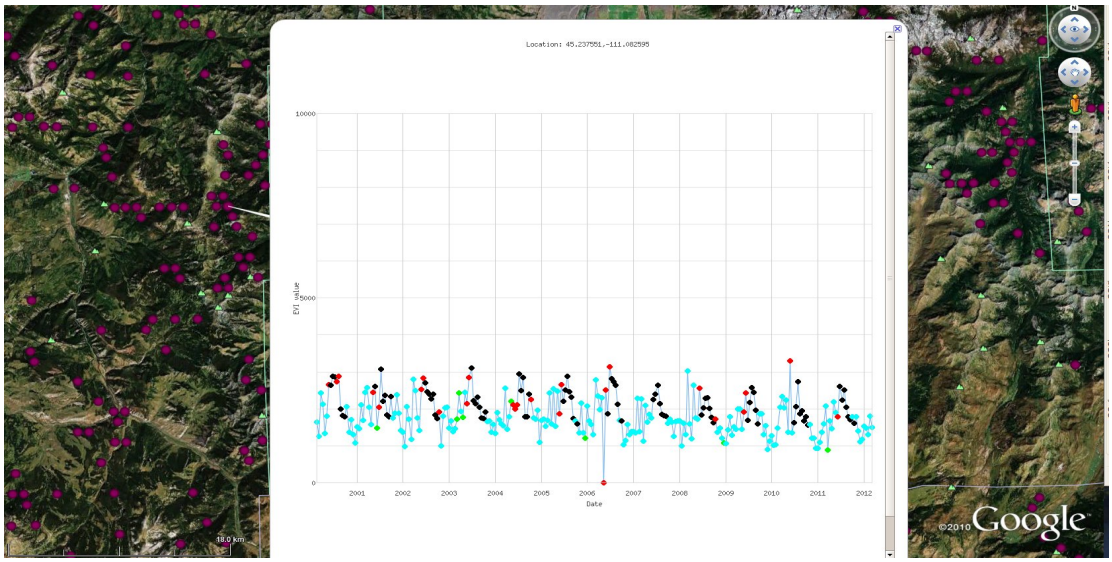


Figure 6.12: A sample insect infested pixel with high rank by PDELTA but low rank by CUSUM.

For our dataset, we determined h as 20,000 and α as 2. Therefore, we compared the pixels that were ranked within top 20,000 by one scheme while less than 60,000 by the other. Figure 6.10 shows the number of pixels lying in each category where we examined the set P and the set C in detail. Figure 6.11 shows the Google Earth snapshot of pixels belonging to set P. These pixels were given high rank by PDELTA and very low rank by CUSUM. On examining majority of these pixels they were found to have very gradual vegetation degradation most likely due to mountain pine beetle infestation (Figure 6.12).

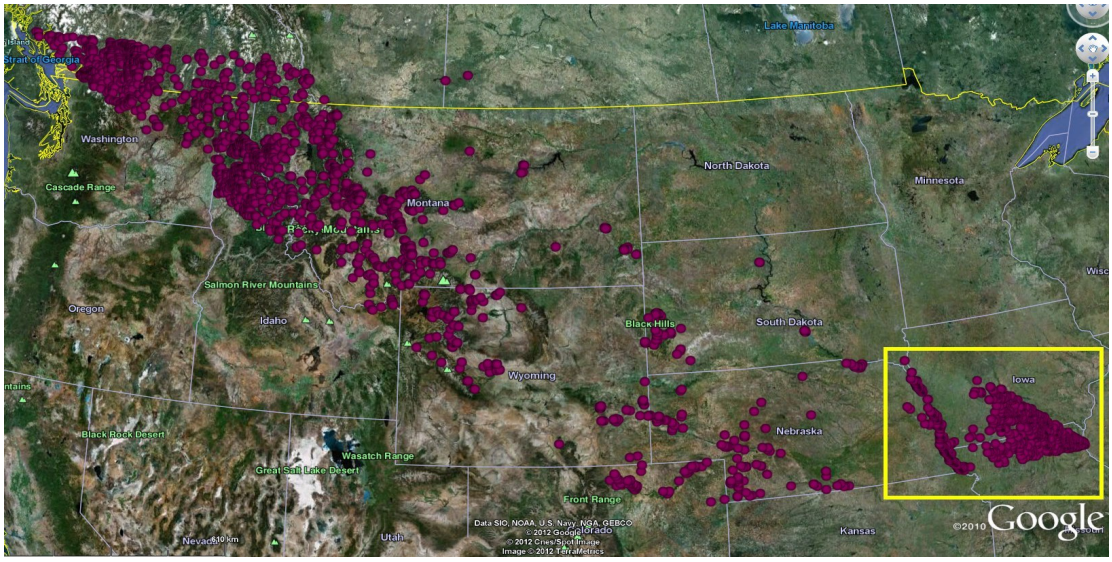


Figure 6.13: Google Earth snapshot of pixels with high scores by CUSUM but low scores by PDELTA. The yellow box denotes farming areas.

Figure 6.13 shows the snapshot of pixels in set C. These are high ranking pixels by CUSUM but very low ranking by PDELTA *change in vegetation distribution* scoring criteria. Most of the pixels contained in this category were found to be farm changes, sudden changes due to fires followed by recovery, unstable gradual changes, or changes occurring towards the end of the time series (boundary conditions). Figure 6.14 shows a farm pixel with vegetation decrease given a high score by CUSUM but not PDELTA. Farming cycles typically have high variability which makes distributions unstable thus resulting a low PDELTA *change in vegetation distribution* score. Excluding farm changes may even be desired because the vegetation changes in farms are

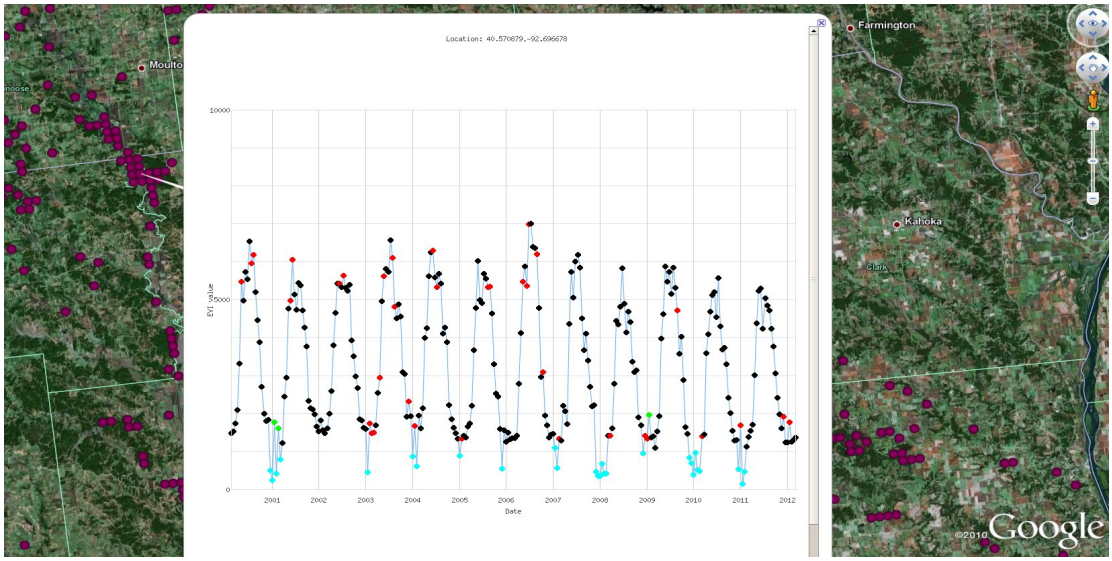


Figure 6.14: A farming change picked up by CUSUM but very low score by PDELTA. Reason: high variability before the start of the change.

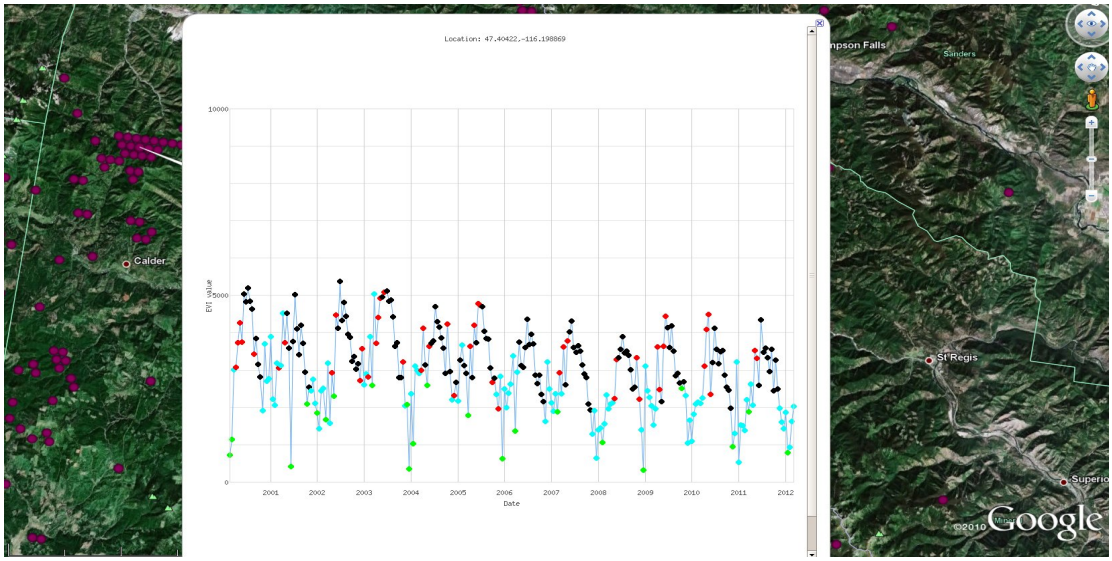


Figure 6.15: Time series given low score by PDELTA *change in vegetation distribution* (rank: 60036) due to high variability before the start of change. NOTE: It was given high score by PDELTA *total loss in EVI* scoring mechanism (rank: 9195).

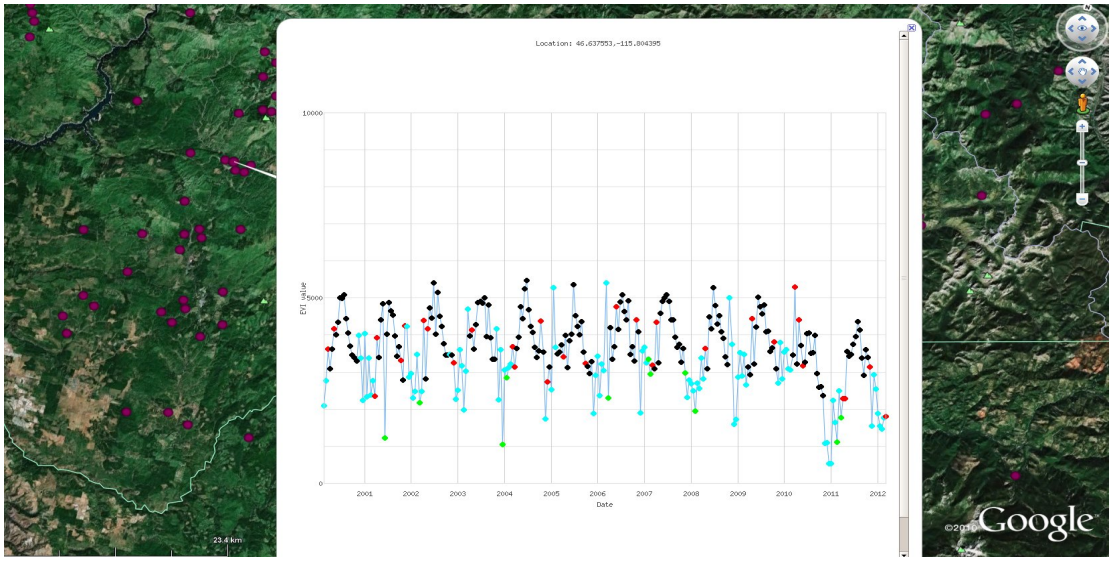


Figure 6.16: Time series given low score by PDELTA *change in vegetation distribution* due to unavailability of data to build the vegetation distribution at the end of the time series. NOTE: It was given high score by PDELTA *Drop* scoring method (rank: 2188).

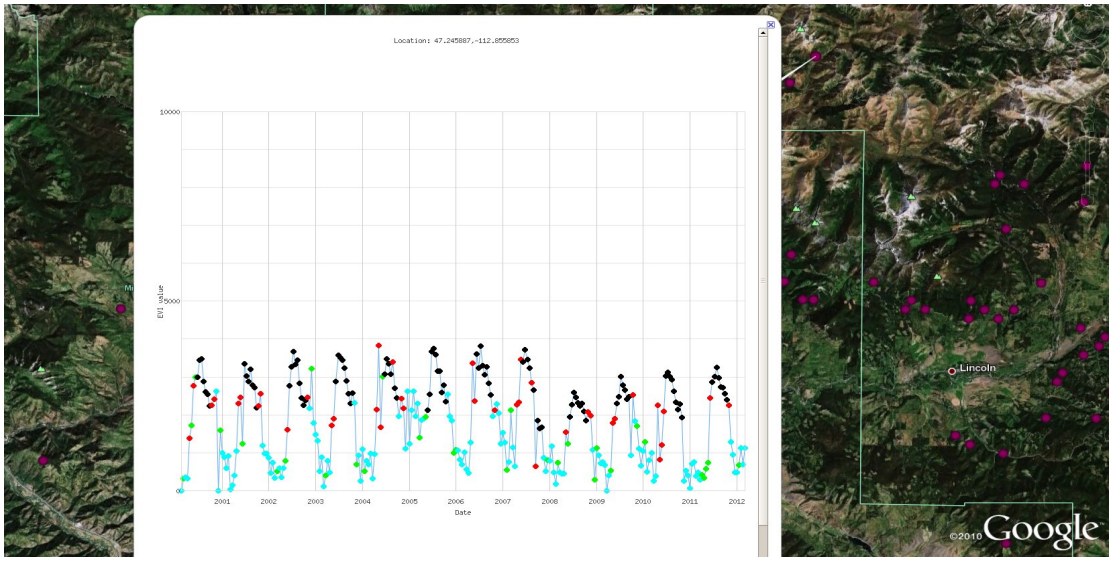


Figure 6.17: Time series given low score by PDELTA *change in vegetation distribution* due to unstable vegetation distribution caused by recovery after fire.

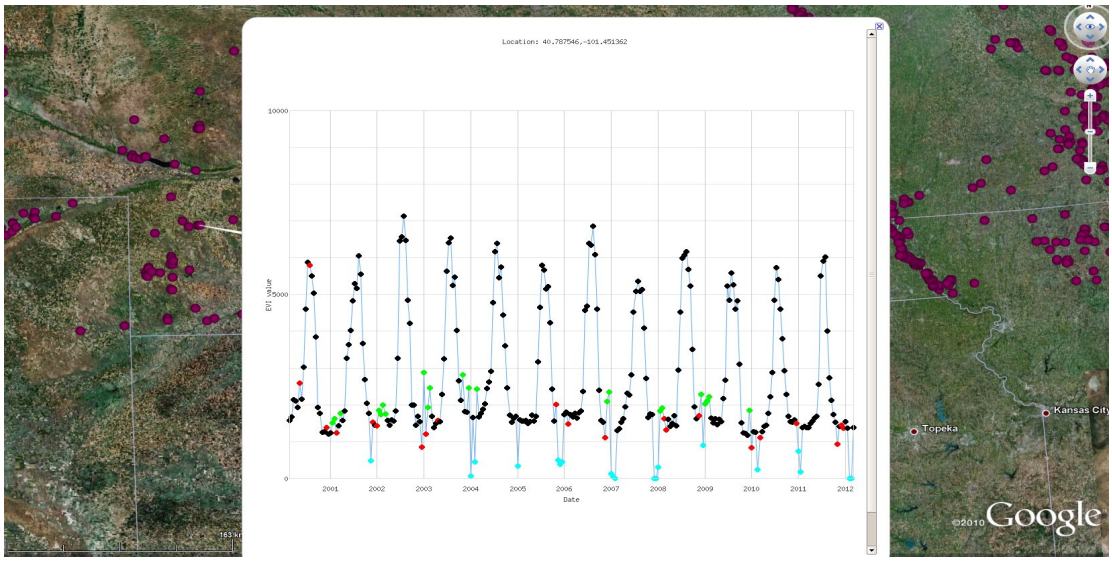


Figure 6.18: Undesired farming time series picked up by CUSUM.

typically not of interest to stake holders who focus mainly on forests. The second type of changes missed by PDELTA were the changes that are desired however due to high variability before the start of the change was given a low score (Figure 6.15). However, this time series was given a high score by PDELTA *total loss in EVI* scoring mechanism. The third type of changes given low score by PDELTA were the boundary cases with insufficient data to build the vegetation distribution after the end of change (Figure 6.16). The fourth type of changes in this category are fire changes followed by recovery which renders the vegetation distribution unstable (Figure 6.17). It should be noted, however, that PDELTA is not intended to find sudden changes as due to fires where recovery is a common phenomenon. Figure 6.18 shows another example of a farming pixel present in this set. It was observed that CUSUM gave high score and detected many such undesired farm pixels, the reason being that average EVI in the first part of the time series was higher than that of the second part.

After qualitatively comparing PDELTA and CUSUM, it was found that CUSUM gave a low score to many gradually decaying insect infested pixels which were successfully given a high score by PDELTA. In addition, CUSUM detected many uninteresting farm pixels. It was also observed that desired changes detected by CUSUM and not by PDELTA *change in vegetation distribution* scoring method were detected using other

PDELTA scoring mechanisms such as *total loss in EVI* or *drop*.

Chapter 7

Discussion & Limitations

Here, we discuss some unresolved issues about the method, suggest some solutions and conclude by some limitations.

7.1 Degree of Suddenness

A major conceptual challenge is the differentiation between a gradual change and other types of changes such as sudden changes. The transition point or period between a sudden change and a gradual one can be highly context dependent and open to individuals interpretation.

Here we try to quantify the suddenness in any drop detected which could be used to systematically eliminate uninteresting changes or the changes deemed not gradual enough. Sudden changes are undesired events in this framework for important reasons such as (1) This approach is designed with focus on detecting gradual changes. Thus its effectiveness in detecting sudden drop cannot be guaranteed. It is undesirable that some sudden changes are detected while others are not. (2) For the purpose of characterization - it is assumed that the events outputted by the overall approach are of gradual nature.

Following two methods are suggested to assess the amount of suddenness during the detected decrease period in a time series.

Percentage decrease at a time instant: This measure is computed as the ratio of the *maximum drop between two consecutive years during the decrease period* and

the *total drop across the decrease period*. If this value is high, it indicates that the majority of the total drop occurred at a single time step, thus giving an indication that the drop is of sudden nature. Similarly, if the value is low it indicates that there is no single time instant at which the drop is comparable to overall drop across the decrease, thus implying a gradual nature of change.

Relative decrease across each time instant: This measure compares the decrease across each time instant relative to the decrease across other time instants during the decrease period. If the decrease across any time instant is significantly larger than the decrease across other time instants, this implies that most of the decrease occurred at a single time step.

7.2 Potential Reliability Condition Augmentation

In section 3.3, we described a method to determine the start and end time steps of a decrease window. As it would appear, these change points are dependent on the reliability condition used. We have described a simple reliability condition that limits the amount of rise in the time series once the drop has begun. If the rise is greater than a threshold, the drop is terminated by positioning a second change point before this rise and a new drop window is initiated at the next candidate start time step.

The advantage of using the above reliability condition is that it is simple, as well as it has only one parameter. But this condition may handle some time series changes differently from what one might prefer. For instance, if there is a time series that undergoes a good drop during the third year, decreases very mildly or not at all for the next five years, and then undergoes a drop again during the eighth and ninth years, then the start and end change points would be identified around the second year and the ninth year respectively. It would overlook that the time series is hardly decreasing for many intermittent years. It might be more desirable to include these two drops in separate drop windows instead of one. But such cases are not a limitation, as the proposed reliability condition can be adapted to handle these cases. It can be taken as a base condition over which other conditions are added, which may or may not be region specific. In the context of the above example problem, one possible adaptation could be to add a condition that the change window must have similar recurrent drops. This

would ensure that a single trend is maintained during a decrease window. Note that this modification would result in the use of auxiliary parameters thus deviating further from simplicity.

7.3 Limitations

- Each scoring mechanism discussed give priority to different types of changes. The amount of variability in vegetation as well changes of interest itself limit the development of a single scoring mechanism to capture all types of changes of interest. However, it was observed that most interesting changes were captured by at least one scoring mechanism.

- None of the scores assign a statistical significance to the changes.

- Small scale changes not apparent in the 1 km or 250 m resolution time series data are not detected. In order to detect such changes a higher spatial resolution data is needed such as 30 meters Landsat image data stack. However, higher resolution spatial data highly restrict scalability of an approach as well as incurs more noise. In addition, Landsat data is not easily available for the whole globe.

- Changes that recover very rapidly, like many deforestation events in the tropical belt, are not detected well. Higher temporal resolution EVI data (1 day temporal resolution) can be used for such a detection, however, it incurs more noise than the 16-day resolution data. Scalability is another issue to deal with higher temporal resolution as a time series would contain 16 times more data.

Part II

Spatio-Temporal Approach - Persistent Spatial Deviance

Chapter 8

Chapter 8

Spatial Analysis

8.1 Motivation

Remote sensing data, especially of vegetation, generally occur in an array of similar time series profiles which may be spread out spatially in a non-contiguous manner. It is easy to imagine this considering similar vegetation grows and is suited to a particular climatic zone. An equally compelling argument is that most vegetation, especially of forested areas, span large expanse of land and thus depict similar vegetation pattern.

Spatial information can be leveraged at either the pre-event detection stage or the post-event detection stage. Pre-event detection is when other pixels in the data space are also used to detect change in the current pixel. The events detected in this manner are result of a spatio-temporal analysis. Post-event detection is when the data space is used to improve the result set after the events are detected using a temporal analysis. This includes using spatial relationships (e.g. proximity) to improve the quality of the result set already obtained or detect new events that were not captured through a purely temporal analysis. In this chapter, we describe a pre-event detection approach (8.2) called *Persistent Spatial Deviance*.

8.2 Persistent Spatial Deviance (PSD)

8.2.1 Key Idea

We propose that under normal (unchanged) circumstance, any given time series, if similar to some other time series for few years, would remain similar to them for the remaining years. Throughout this chapter, the term *neighbor* implies time series with similar temporal profile to the given time series of interest and does not mean neighbor geo-spatially. If a time series is similar to some other time series for few years but persistently deviates from them after a certain time step, it indicates a likelihood of vegetation change. This persistent deviation of any time series with respect to its neighbors can be quantified as follows. First, those time series are identified which, for few initial years, depict a similar time series pattern with respect to the given time series of interest. This can be achieved using a similarity measure (e.g. City-block distance) and a similarity threshold. These newly identified similar time series forms the initial neighborhood or the neighborhood model. After the initial neighborhood is constructed, separate vegetation distributions are examined for each time step following the initial years for which the neighborhood model was built. Key idea here is that for each time step if the vegetation level of the given time series manifest as anomaly in distribution of vegetation levels of its *unchanged neighbors*, the given time series is flagged as changed.

Using space in this manner can detect changes that cannot be detected using a purely temporal analysis. An example is a highly variable time series with insignificant amount of decrease after a certain time step with respect to its temporal history. However, this insignificant decrease from its own temporal history can at the same time be a significant decrease from its unchanged neighborhood.

8.2.2 Unchanged Neighborhood

When the initial neighborhood is built for a certain time duration, we know that all neighbors are similar to the given time series for that time duration. But the behavior of the initial neighbors and the given time series is not known during the following time steps. For the following time steps, we not only want to determine the deviation of the current time series from its neighbors, but we also want to ensure that at each of the following time step we determine the deviation from only the unchanged neighbors at that

time step. If neighborhood pixels have also undergone change, the current time series may not appear as an anomaly with respect to its neighbors. Therefore, our immediate objective is to identify those neighbors at any given time step that are unchanged. In order to accomplish this objective, we construct the unchanged neighborhood under the following assumptions which should hold for any given land cover type.

Assumptions:

1. *At any time step, the majority of initial neighborhood pixels remain unchanged with values having higher density than the changed pixels.* To ensure this, two considerations are made while constructing the initial neighborhood. 1) The physical space near the time series of interest is not searched for similar time series. The reason being that land cover events are highly spatially auto-correlated meaning that nearby locations are likely to have undergone the similar change as the time series of interest. This would result in pumping more changed time series in the initial neighborhood that we want to avoid. Since most land covers span large areas of land, it is preferable to identify similar time series from far off locations which are less likely to be affected by the change event. 2) For the given time series, the initial neighborhood is kept updated till before a potential change time step. This avoids changes in the initial neighborhood pixels between when the neighborhood model was built and when the change started in the time series of interest.

2. *At any time step, the vegetation observations of the unchanged pixels follow normal distribution.*

3. *At any time step, the pixels that are changed are uni-directional, i.e. the vegetation either decrease or increase but not both.*

For most practical purposes, these assumptions are reasonable.

In order to separate the unchanged vegetation observations from the changed ones, the following steps are performed on the initial neighborhood pixels *for each time step* following the time steps on which the initial neighborhood was built.

First The kernel density of vegetation observations is estimated and the location of the highest peak of this density curve is identified. According to our assumptions, the unchanged distribution should be densest and therefore this peak would lie at the expected mean of the distribution of unchanged vegetation values.

Second Since the change is uni-directional, the unchanged vegetation values follow normal distribution on that side of the expected mean which is away from the change. We identify that side as the one having a lower standard deviation (since the standard deviation would be larger in the direction of changed values).

Using the above two steps, we get the mean and the standard deviation of the vegetation values of the unchanged neighbors at each time step. Figure 8.1 shows the upper and lower standard deviation bounds on the unchanged neighborhood of a given time series.

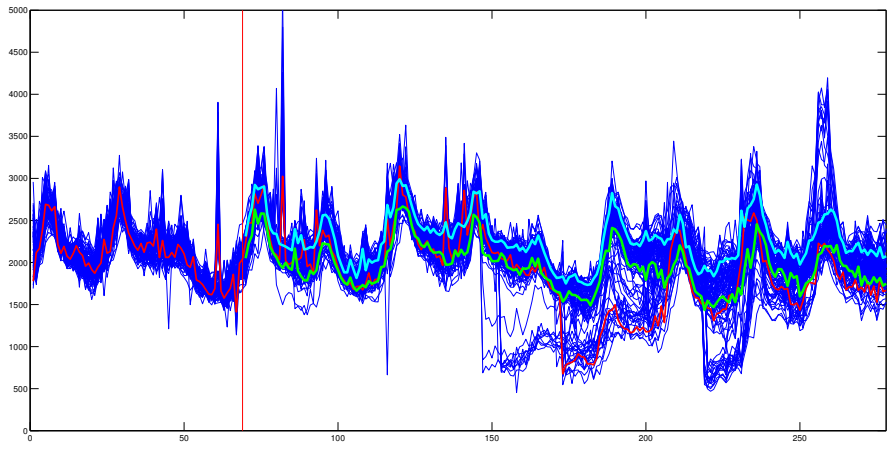


Figure 8.1: A given time series which is to be verified as changed or unchanged (red), similar neighborhood (blue), plus and minus one standard deviation bounds of the unchanged vegetation values (cyan and green respectively).

8.2.3 Change Score & Change Time

The change score for a given time series T is based on the persistence of the deviation of the time series T from its unchanged neighbors. For the time series T with n observations, each time step t_i is picked one at a time as a potential change point. The initial neighborhood is built on its time series pattern during the immediate p previous observations of t_i. Since until time step t_i, the time series T is similar to its neighbors (by definition), the deviation score or change score for the time series T at time step t_i is computed on the time steps following time step t_i. At each time step in $[t_{i+1}, t_n]$, the vegetation value in time series T is observed against its corresponding unchanged neighbor's vegetation value distribution. The deviation score for time series T is calculated as follows: Beginning with time step t_{i+1}, at each time step the *distance* (with sign) of the vegetation level of time series T from the expected mean of the corresponding-time-step vegetation distribution of the unchanged neighbors is noted. The *distance* measure can be either of the following:

- Standard deviation units (or z-score), i.e. how many standard deviations away is the vegetation level of time series T from the expected mean of its corresponding distribution.
- Simple integral difference, i.e. the difference in EVI values of the vegetation level of time series T and the expected mean EVI value of its corresponding distribution. **NOTE:** *For this distance measure, Assumption 3 stated above is not required since the standard deviation bounds on the unchanged neighborhood is not needed.*

A cumulative sum of this *distance* from time step t_{i+1} to t_n is maintained. Note that the cumulative sum will keep aggregating a positive value while the current time series T is persistently deviant from its unchanged neighbors in a specific direction. If and as soon as T recovers from the change and its temporal profile becomes similar to its unchanged neighbors, the score would reach its expected peak. The change score for time step t_i, S_{t_i}, is the maximum value of this running sum. This whole process is repeated for each t_i in time series T, $\forall i \in [p, n]$.

The most likely time step of change is the one for which S_{t_i} is maximum. This time step is termed as the start time step, t_α, and the corresponding score is deemed

as the change score of the time series T, S_T. The corresponding stop time step, t_β, is defined as the time step when the change is recovered and the time series T pattern again becomes similar to its unchanged neighbors (or end of the time series, which ever is earlier).

8.3 Discussion

This approach is robust in detecting the unchanged neighborhood inspite of having a large number of changed time series in the initial neighborhood (Figure 8.2), as long as the unchanged vegetation values have highest density. In addition, outliers in the neighborhood time series has little or no adverse effect on correctly detecting the unchanged neighborhood (see time steps 250 through 265 in Figure 8.1 for clarification). Even if the outliers distort the unchanged neighborhood for few time steps, the scoring is based on the persistence of deviation in a specific direction and the presence of outliers at few time steps would have insignificant contribution to the aggregate score.

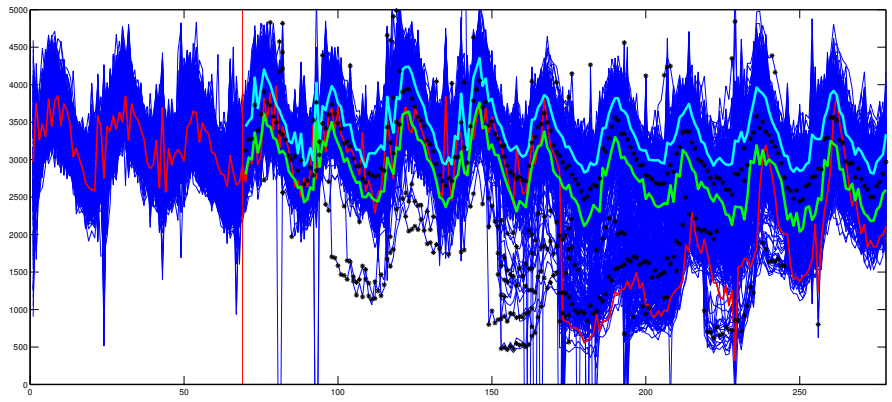


Figure 8.2: Unchanged neighborhood built correctly despite having a large number of changes in the initial neighborhood. A given time series which is to be verified as changed or unchanged (red), similar neighborhood (blue), plus and minus one standard deviation bounds of the unchanged vegetation values (cyan and green respectively), kernel density modes at each time step (black stars).

Also note that Assumption 1 is important for correctly detecting a time series as changed. By building the initial neighborhood cautiously as described in Assumption 1, it is unlikely that this assumption would be violated. However still, we do a diagnostic test to ensure that the unchanged neighborhood is built correctly and not something like as in Figure 8.3 where cautions in Assumption 1 were not heeded. The main problem would occur when at any time step the density of the unchanged vegetation values is less than that of changed (or other) vegetation values. To detect such a situation, a comparison between the few previous-time-step expected means of unchanged vegetation and the current-time-step expected mean of the unchanged vegetation is made. If the current-time-step expected mean of the unchanged vegetation is similar to the previous expected means, then the current expected mean is more likely to be correctly detected as that of the unchanged vegetation.

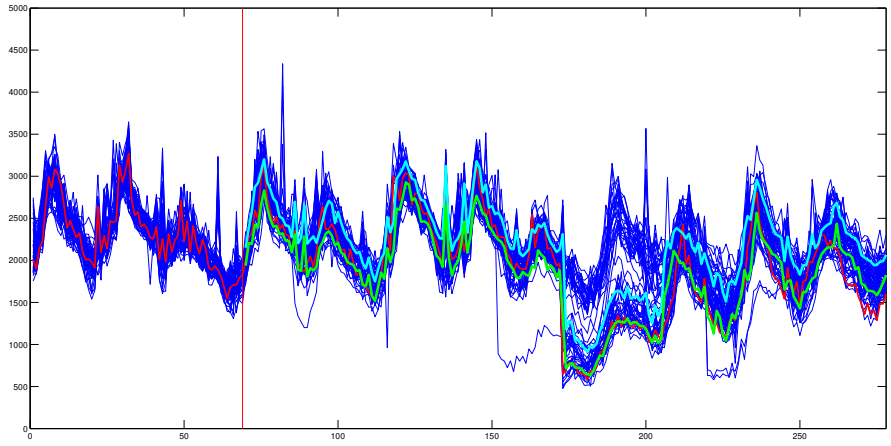


Figure 8.3: Unchanged neighborhood built incorrectly after approximate time step 170. A given time series which is to be verified as changed or unchanged (red), similar neighborhood (blue), plus and minus one standard deviation bounds of the unchanged vegetation values (cyan and green respectively).

8.4 Limitations

- One of the major limitations of this approach is its time complexity. The first part is to build the initial neighborhood for each time series to be verified as changed or unchanged. For this, a similarity query is performed on the given data set. This step can become very time consuming if either the number of time series or the number of observations in a time series is large. To speed up the time series query process, a fast approach for similarity query is needed. One such spatio-temporal range query approach is presented in appendix A which identifies the similar time series using lesser number of comparisons than using an exhaustive search.

- Another limitation is the non-guarantee that unchanged neighborhood is densest. Though we have taken steps to avoid such a scenario, this can still occur in an under-represented land cover type most of which has undergone change.

Chapter 9

Concluding Remarks and Direction for Future Work

9.1 Summary

In this thesis, we presented two approaches to detect vegetation changes.

First, we presented a globally scalable approach, PDELTA (I), for detecting a gradually decreasing EVI time series that can capture changes caused by a variety of sources. PDELTA can be considered an adaptation of CUSUM with the added capability of identifying the period of decrease and quantifying the magnitude of change in a time series, while being more robust in the presence of noise and spurious changes. We also described five scoring mechanisms to capture different types of changes. Importantly, these scoring mechanisms can be used for change characterization as they focus on different characteristics of the change. We demonstrated the efficacy of the proposed approach using independent validation data sets in Colorado and Madagascar. It was also shown that genuine changes were detected by our technique which were missed by other approaches, as well as points identified as changed by other approaches with no perceptible EVI signal were not detected. We quantitatively evaluated our technique with CUSUM, and the state of the art BFAST technique. BFAST in its present form is computationally very expensive, whereas both PDELTA and CUSUM are quite fast. PDELTA can also identify reforested areas depicted by increase in vegetation simply by reversing a time series before applying this algorithm. We presented qualitative case

studies on the application of PDELTA to detect insect infested areas in British Columbia and Montana as well as logging in Peru. Lastly, we proposed a new qualitative evaluation methodology and performed comparison between PDELTA and CUSUM methods. We concluded with some limitations of the current framework.

Second, we incorporated the concept of *persistent change* into a spatio-temporal framework and presented another approach, PSD (II), to detect more general types of vegetation changes including abrupt changes. This approach was not evaluated extensively due to time limitations. Some qualitative evaluations of this approach highlight its potential in detecting changes which may not be detected using a purely temporal approach. We concluded with discussion and limitations of this framework.

9.2 Future Work

- In the current study, we did not assign a statistical significance to any scoring method discussed. Statistical significance to scores and thus changes can also be used for automatic thresholding for differentiating between changed and unchanged pixels. Such a formulation in future would be a good addition to these approaches.

- PDELTA approach to detect multiple decreasing segments in a time series has found application in detecting ocean eddies using sea surface height data [19]. This direction is being further explored.

- The limitations of both the approaches include the inability to detect small scale changes not evident in the EVI time series. For detecting such small scale changes, a classification model can be built for the known EVI changes using the reflectance bands of higher resolution Landsat data. This classification model can then be used to correctly classify changed regions that are not identifiable using EVI signal.

References

[1] P. Coppin, I. Jonckheere, K. Nackaerts, B. Muys, and E. Lambin. Digital change detection methods in ecosystem monitoring: a review. *International Journal of Remote Sensing*, 25(9):1565–1596, 2004.

[2] D. Lu, P. Mausel, E. Brondízio, and E. Moran. Change detection techniques. *International Journal of Remote Sensing*, 25(12):2365–2401, 2003.

[3] J.E. Vogelmann, G. Xian, C. Homer, and B. Tolk. Monitoring gradual ecosystem change using landsat time series analyses: Case studies in selected forest and rangeland ecosystems. *Remote Sensing of Environment*, 2012.

[4] Yashu Chamber, Ashish Garg, Varun Mithal, Ivan Brugere, Vipin Kumar, Michael Lau, Michael Steinbach, Christopher Potter, Steven A. Klooster, Vikrant Krishna, and Shyam Boriah. A novel time series based approach to detect gradual vegetation changes in forests. In *CIDU*, pages 248–262, 2011.

[5] L. Eklundh, T. Johansson, and S. Solberg. Mapping insect defoliation in Scots pine with MODIS time-series data. *Remote sensing of Environment*, 113(7):1566–1573, 2009.

[6] V. Mithal, A. Garg, S. Boriah, M. Steinbach, V. Kumar, C. Potter, S. Klooster, and J.C. Castilla-Rubio. Monitoring global forest cover using data mining. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(4):36, 2011.

[7] J. Kucera, P. Barbosa, and P. Strobl. Cumulative sum charts-a novel technique for processing daily time series of modis data for burnt area mapping in Portugal.

In *MultiTemp 2007: International Workshop on the Analysis of Multi-temporal Remote Sensing Images*, pages 1–6. IEEE, 2007.

[8] J. Verbesselt, R. Hyndman, G. Newnham, and D. Culvenor. Detecting trend and seasonal changes in satellite image time series. *Remote sensing of Environment*, 114(1):106–115, 2010.

[9] R.E. Kennedy, Z. Yang, and W.B. Cohen. Detecting trends in forest disturbance and recovery using yearly landsat time series: 1. landtrendr–temporal segmentation algorithms. *Remote Sensing of Environment*, 114(12):2897–2910, 2010.

[10] Land Processes Distributed Active Archive Center. `http://edcdaac.usgs.gov`.

[11] Sophocles J. Orfanidis. *Introduction to signal processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, 1995.

[12] Giles M. Foody. Status of land cover classification accuracy assessment. *Remote Sensing of Environment*, 80(1):185–201, 2002.

[13] E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

[14] Shyam Boriah. *Time Series Change Detection: Algorithms for Land Cover Change*. PhD thesis, University of Minnesota, 2010.

[15] U.S. Department of Agriculture, Forest Service, Rocky Mountain Region, Forest Health Management. `http://www.fs.fed.us/r2/fhm`.

[16] Russell T. Graham. Hayman fire case study. Technical Report RMRS-GTR-114, U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station, 2003.

[17] Grady J. Harper, Marc K. Steininger, Compton J. Tucker, Daniel Juhn, and Frank Hawkins. Fifty years of deforestation and forest fragmentation in madagascar. *Environmental Conservation*, 34(4):325–333, 2007.

[18] G.P. Asner, G.V.N. Powell, J. Mascaro, D.E. Knapp, J.K. Clark, J. Jacobson, T. Kennedy-Bowdoin, A. Balaji, G. Paez-Acosta, E. Victoria, et al. High-resolution

forest carbon stocks and emissions in the amazon. *Proceedings of the National Academy of Sciences*, 107(38):16738–16742, 2010.

[19] J.H. Faghmous, Y. Chamber, S. Boriah, S. Liess, V. Kumar, F. Vikebø, and M. dos Sontos Mesquita. A novel and scalable spatio-temporal technique for ocean eddy monitoring. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[20] Pusheng Zhang, Shashi Shekhar, Y. Huang, and Vipin Kumar. Spatial cone tree: An index structure for correlation-based similarity queries on spatial time series data. In *The Proc. of the International Workshop on Next Generation Geospatial Information*. Citeseer, 2003.

[21] Hanan Samet. The quadtree and related hierarchical data structures. *ACM Computing Surveys (CSUR)*, 16(2):187–260, June 1984.

[22] R.H. Güting. An introduction to spatial database systems. *The VLDB Journal*, 3(4):357–399, 1994.

[23] Volker Gaede and O. Günther. Multidimensional access methods. *ACM Computing Surveys*, 30(2):170–231, 1998.

[24] A. Guttman. R-trees: A dynamic index structure for spatial searching. *ACM SIGMOD Record*, 1984.

[25] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The R*-tree: an efficient and robust access method for points and rectangles. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 19(2):322–331, June 1990.

[26] A. Camerra, T. Palpanas, J. Shieh, and E. Keogh. isax 2.0: Indexing and mining one billion time series. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 58–67. IEEE, 2010.

[27] V. Niennattrakul, P. Ruengronghirunya, and C.A. Ratanamahatana. Exact indexing for massive time series databases under time warping distance. *Data Mining and Knowledge Discovery*, 21(3):509–541, 2010.

[28] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. Locally adaptive dimensionality reduction for indexing large time series databases. *ACM SIGMOD Record*, 30(2):151–162, 2001.

[29] Jin Shieh and Eamonn Keogh. iSAX: disk-aware mining and indexing of massive time series datasets. *Data Mining and Knowledge Discovery*, 19(1):24–57, Feb 2009.

[30] Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. Fast subsequence matching in time-series databases. *ACM SIGMOD Record*, 23(2):419–429, 1994.

[31] K.P. Chan and A.W.C. Fu. Efficient time series matching by wavelets. In *ICDE'99: Proceedings of 15th International Conference on Data Engineering*, pages 126–133. IEEE, 1999.

[32] E. Keogh and M. Pazzani. A simple dimensionality reduction technique for fast similarity search in large time series databases. *Knowledge Discovery and Data Mining. Current Issues and New Applications*, pages 122–133, 2000.

[33] Eamonn Keogh, Kaushik Chakrabarti, and Michael Pazzani. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, 3(3):263–286, August 2001.

[34] B.K. Yi and C. Faloutsos. Fast time sequence indexing for arbitrary Lp norms. In *VLDB '00: Proceedings of the 26th International Conference on Very Large Data Bases*, pages 385–394, 2000.

[35] Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi. Experiencing SAX: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15(2):107–144, April 2007.

[36] Donald J. Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD Workshop*, pages 359–370, 1994.

[37] MF Mokbel and TM Ghanem. Spatio-temporal access methods. *IEEE Data Engineering*, pages 1–10, 2003.

[38] Long-Van Nguyen-Dinh, Walid G. Aref, and Mohamed F. Mokbel. Spatio-temporal access methods: Part 2 (2003 - 2010). *IEEE Data Eng. Bull*, 33(2):46–55, 2010.

[39] Panfeng Zhou, Donghui Zhang, Betty Salzberg, Gene Cooperman, and George Kollios. Close pair queries in moving object databases. In *Proceedings of the 13th Annual ACM International Workshop on Geographic Information Systems*, pages 2–11, 2005.

[40] Jinfeng Ni and C. Ravishankar. PA-tree: A parametric indexing scheme for spatio-temporal trajectories. *Advances in Spatial and Temporal Databases*, pages 923–923, 2005.

[41] Jinfeng Ni and C.V. Ravishankar. Indexing spatio-temporal trajectories with efficient polynomial approximations. *IEEE Transactions on Knowledge and Data Engineering*, 19(5):663–678, May 2007.

[42] Kostas Patroumpas and Timos Sellis. Monitoring orientation of moving objects around focal points. *Advances in Spatial and Temporal Databases*, pages 228–246, 2009.

[43] Longhao Wang, Yu Zheng, Xing Xie, and W.Y. Ma. A flexible spatio-temporal indexing scheme for large-scale GPS track retrieval. In *MDM'08: Proceedings of the 9th International Conference on Mobile Data Management*, pages 1–8. IEEE, April 2008.

[44] J.M. Patel, Yun Chen, and V.P. Chakka. STRIPES: an efficient index for predicted trajectories. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, pages 635–646, 2004.

[45] S. Saltenis, C.S. Jensen, S.T. Leutenegger, and M.A. Lopez. *Indexing the positions of continuously moving objects*, volume 29. ACM, June 2000.

[46] Yufei Tao, D. Papadias, and Jimeng Sun. The TPR*-tree: an optimized spatio-temporal access method for predictive queries. In *VLDB'03: Proceedings of the 29th International Conference on Very Large Data Bases*, volume 29, pages 790–801, 2003.

Appendix A

Spatio-temporal Indexing Structure

A.1 Introduction

Similarity search is a fundamental operation in exploratory data analysis. Sequential scan is prohibitively expensive for datasets that do not fit into memory as it would require many disk accesses. Besides, it is also slow for datasets consuming most of the memory. There are many potential use cases for similarity queries on Earth science data. For example, finding interesting patterns in vegetation data could reveal regional phenomena such as pest infestations or droughts, clustering methods use similarity search to group time series with minimal loss of information, and similarity search can find surrogate time series for the identification and imputation of noisy values. In order to fully address these key applications, the spatial aspect of the data must be combined with temporal aspect and be incorporated into the indexing structure to allow for spatially-constrained queries, which reduce the query time and limit the result set to a specified spatial region or set of regions. [20] described one such indexing structure, called Spatial Cone Tree, which forms an index tree by recursively partitioning a time series dataset into fixed, equi-sized spatial regions.

In this appendix, we introduce a novel method for efficiently indexing and searching spatio-temporal data. The proposed approach is a two step adaptation of the spatio-temporal indexing structure presented in [20]. [20] described a method to query a

given time series based on its correlation with other time series. All time series with sufficient correlation, a user-defined parameter, with the query time series are identified. However, this approach has two limitations for our problem. First, correlation is not a good similarity measure for many spatio-temporal data, e.g. vegetation data, where the absolute values matter in addition to the seasonal trends. Second, the splitting at each node in the index tree is based on a simple quad tree approach [21], which recursively divides the space represented by a tree node into four equally sized regions. This type of spatial partitioning may work for data of climatic variables with smooth spatial boundaries such as Sea Level Pressure or Land Surface Temperature. However, for vegetation type data, the space should be carefully partitioned taking into account the characteristics of time series in each partition.

We adapt the Spatial Cone Tree to a new indexing structure, *Spatial Containers*, aimed at addressing the above two issues. First, we adapt the correlation measure to City-block distance (or L_1-norm), or in general the Minkowski distance (L_p-norm), which is a more suitable measure of similarity for vegetation type data. Second, instead of recursively dividing the space into four equally sized regions, we suggest a *Spatially Constrained Partitioning* where the space division is data driven to minimize the information overlap between different partitions.

Section A.2 discusses existing indexing techniques for spatial, temporal and spatio-temporal datasets. Section A.3 summarizes the basic concepts behind the spatial cone tree. Sections A.4,A.5,A.6 describe the proposed spatio-temporal indexing structure.

A.2 Related Work

Spatial databases, which support queries on data types such as points, lines, and polygons, have been evolving for a number of years now [22]. Spatial access methods [23] are broadly grouped into two categories: space-driven structures derived from the quad tree [21], which use a multi-level grid to hierarchically partition space; and data-driven structures such as the R-tree [24] and R*-tree [25], which group a set of spatial objects using hierarchical, overlapping minimum bounding rectangles. These methods can efficiently query spatial datasets on the order of billions of points but these methods do not bring time into consideration.

Similar advances have been made in scalable time series indexing [26, 27], which generally relies on two considerations. The first is the data representation and may involve approximations with properties such as lower-bound to a particular distance (e.g., Euclidean [28, 29]); common representations use discrete Fourier transforms [30], wavelets [31], piecewise linear approximation [32, 33], and piecewise aggregate approximation [34, 35]. The second is the similarity measure; examples include Manhattan, Euclidean, correlation, and dynamic time warping (DTW) [36].

Moderate work exists in the area of spatio-temporal indexing. Several methods for point and moving object data (trajectories) are found in the literature; [37] and [38] summarize more than ten such tree structures and derivatives, including 3-D structures [39], overlapping or multi-version structures [40, 41], and specialized trajectory structures [42, 43]. Methods based on quad-trees [44] and the R-tree family [45, 46] have also been adapted for trajectories.

A.3 Key Concepts

The access structure proposed in this appendix is based on the concept similar to that of the Spatial Cone Tree. The spatial cone tree is a tree of cones (or nodes) with each node representing a number of time series. The term 'cone' is used because each time series in a node is normalized and lie on the surface of a multi-dimensional unit sphere and has the following properties: *axis*, which is the mean of all normalized time series in the cone, and *span*, which is the maximal angle between any normalized time series in the cone and the axis of the cone. The number of time series in the cone, N, is also noted. The root node of the tree represent all the time series in the dataset. Beginning with the root node, the set of time series in each node is recursively and exhaustively partitioned into child nodes until the span of the nodes (leaf nodes) and the number of time series in each leaf become less than pre-specified thresholds max_span and M respectively. The recursive partitioning is done based on the Simple QuadTree method [21].

The spatial cone tree uses correlation as the similarity measure and amongst other operations support efficiently identifying all time series that are sufficiently correlated with the query time series given a correlation threshold. It uses a filter and refine

approach to identify all similar time series. The filtering step involves traversing the index tree starting at the root node. At each node it determines the relationship between that node and the query time series as one of the following: all-true (the whole node and therefore its children are sufficiently similar to the query), all-false (the whole node is sufficiently dissimilar to the query) and some-true. Traversing deeper is required only at the some-true nodes. If a leaf node is a some-true node, a refinement step is performed which manually checks the similarity between the query and the time series in that leaf.

A.4 Proposed Approach

The proposed indexing structure is based on the similar concepts defining the spatial cone tree, and therefore a number of definitions described in this section bear a resemblance to the ones presented in the previous section. The theory, however, proposed in our approach is different and is described below.

A spatial container represents a collection of time series. The similarity between two time series is based on the distance as defined by L_p-norm distance measure. Normalization of time series is not required, as is the case with the cone tree. Each spatial container has the following attributes:

1. Size: The number of time series being represented.

2. Axis: The mean of all time series within a container.

3. Span: The maximum distance from any time series in the container to the Axis.

Starting from a single container representing the entire spatio-temporal dataset, a container is recursively split if any of the following conditions are not met.

1. The Size of the container is smaller than a specified number M.

2. The Span of the container is less than a specified value.

This process will produce a tree of containers, where all leaf nodes satisfy the above conditions while all internal nodes do not.

Container Properties

Let a time series T is denoted by:

$$v_1, v_2, v_3, ..., v_n$$

where n are the number of observations and $v_1, v_2, v_3, ..., v_n$ are the values at successive time steps $t_1, t_2, ..., t_n$ respectively.

Axis (A)

Let the axis A be a vector given by $a_1, a_2, ...a_n$ which is the average of all the time series in a container $\mathbb{C}$. Therefore, if $T^1, T^2, ..., T^K$ are K time series in $\mathbb{C}$, then

$$a_i = \frac{{v^1}_i + {v^2}_i + ... + {v^K}_i}{K}$$

Span (S)

The *Span* is given by

$S = max\{L_p(T^i, A)\}$

where $L_p(X, Y)$ denotes L_p-norm distance between the time series vectors X and Y.

A.5 Index Tree Construction

A container (tree node) is recursively split until both of the following conditions are satisfied by a node.

1. The *Size* of a container is smaller than a specified number M.
2. The *Span* of a container is less than a specified value ϵ.

Given M and ϵ, in order to construct a smaller tree (and thus a faster search) node splitting should be such that similar time series are grouped together in the same node.

[20] used a simple quad tree approach for splitting which splits the spatial data (represented by a node) into four equal regions. This splitting works well on Sea Surface Pressure type data which depicts very high spatial autocorrelation with smooth transition boundaries. Though vegetation also show high autocorrelation, similar vegetation can be present in non-contiguous spatial locations. In addition, the transition boundaries can be abrupt with disparate vegetation time series profiles being adjacent to one another resulting in high vegetation diversity in a given contiguous spatial location.

For the above reasons, a fast spatial clustering approach would be much suitable and is used for data space partitioning. This spatial clustering algorithm is repeatedly applied to split each node into k children until the above conditions are met.

A.6 Querying

In the previous section, we described the construction of the index tree. Here, we will derive conditions tested at each node which guides the given query through the index tree. Based on these conditions, the decision regarding whether to enter the subtree of a node is made. Traversing through each node requires a single comparison with the *axis* of that node, resulting in a faster search. If the axis is *sufficiently* similarity to the query, then all time series represented by that node (and therefore its subtree) are similar to the query. Likewise. if the axis is sufficiently dissimilar from the query, all time series represented by that node (and its children) are dissimilar to the query.

The formulation of the sufficient conditions depend on an important convex property applicable to a collection of time series. This property states that given a set of time series, the distance between any two time series is less than or equal to twice the distance between the *mean* time series and the time series farthest from the mean. This phenomenon is easy to visualize in a two-dimensional space (a time series with two observations) using Manhattan or Euclidean distances. However, we prove below that this is true for any number of dimensions and any L_p norm $\forall p \in 1, 2, \cdots \infty$.

A.6.1 Time Series Convex Property

Theorem 1: The Minkowski distance (L_p-norm) between any two time series within a node is less than or equal to twice the distance between the axis and the time series

farthest from the axis.

Input: A given node where A be the *axis*, Z be the time series farthest from the axis, U and V are any two time series within the node. Let,

$A = a_1 a_2 a_3 \cdot a_n$
$Z = z_1 z_2 z_3 \cdot z_n$
$U = u_1 u_2 u_3 \cdot u_n$
$V = v_1 v_2 v_3 \cdot v_n$

To Prove:

$$2 * \sqrt[p]{\sum_{i=1}^{n} |z_i - a_i|^p} \geq \sqrt[p]{\sum_{i=1}^{n} |u_i - v_i|^p} \tag{A.1}$$

Proof: Since Z is the time series farthest from A, therefore the distance between A and Z is more than the distance between A and U or the distance between A and V.

$$\sqrt[p]{\sum_{i=1}^{n} |z_i - a_i|^p} \geq \sqrt[p]{\sum_{i=1}^{n} |u_i - a_i|^p} \tag{A.2}$$

$$\sqrt[p]{\sum_{i=1}^{n} |z_i - a_i|^p} \geq \sqrt[p]{\sum_{i=1}^{n} |v_i - a_i|^p} \tag{A.3}$$

Adding equations A.2 and A.3 gives

$$\implies 2 * \sqrt[p]{\sum_{i=1}^{n} |z_i - a_i|^p} \geq \sqrt[p]{\sum_{i=1}^{n} |v_i - a_i|^p} + \sqrt[p]{\sum_{i=1}^{n} |u_i - a_i|^p} \tag{A.4}$$

From Minkowski Inequality, for any two vectors X and Y, such that

$X = x_1 x_2 x_3 \cdot x_n$
$Y = y_1 y_2 y_3 \cdot y_n$

the following holds:

$$\sqrt[p]{\sum_{i=1}^{n}|x_i+y_i|^p} \leq \sqrt[p]{\sum_{i=1}^{n}|x_i|^p} + \sqrt[p]{\sum_{i=1}^{n}|y_i|^p} \tag{A.5}$$

Let $x_i = x_i - a_i$ & $y_i = a_i - y_i$

$$\implies \sqrt[p]{\sum_{i=1}^{n}|(x_i-a_i)+(a_i-y_i)|^p} \leq \sqrt[p]{\sum_{i=1}^{n}|x_i-a_i|^p} + \sqrt[p]{\sum_{i=1}^{n}|a_i-y_i|^p}$$

$$\implies \sqrt[p]{\sum_{i=1}^{n}|(x_i-y_i)|^p} \leq \sqrt[p]{\sum_{i=1}^{n}|x_i-a_i|^p} + \sqrt[p]{\sum_{i=1}^{n}|y_i-a_i|^p}$$

$$\implies \sqrt[p]{\sum_{i=1}^{n}|(u_i-v_i)|^p} \leq \sqrt[p]{\sum_{i=1}^{n}|u_i-a_i|^p} + \sqrt[p]{\sum_{i=1}^{n}|v_i-a_i|^p} \tag{A.6}$$

Equations A.4 and A.6 gives

$$2 * \sqrt[p]{\sum_{i=1}^{n}|z_i-a_i|^p} \geq \sqrt[p]{\sum_{i=1}^{n}|(u_i-v_i)|^p} \tag{A.7}$$

$$= \quad equation \quad A.1$$

Hence Proved.

A.6.2 Sufficient Conditions

Let δ be the distance threshold for the given query time series Q. At each node, we compute the L_p distance between the *axis* and Q. Let this distance be denoted by λ. Depending on the distance, one of the following conditions is true:

1. $\lambda + Span \leq \delta$

   Retrieve all time series from the container. (ALL TRUE)

2. $\lambda - Span > \delta$

   Don't retrieve any time series from the container. (ALL FALSE)

3. *Otherwise*

   Analyze the children nodes. (SOME TRUE)

Note that these conditions are valid because the time series convex property A.6.1 holds.