Classifying Multivariate Time Series by Learning Sequence-level Discriminative Patterns

Guruprasad Nayak[1] Varun Mithal[2] Xiaowei Jia[1] Vipin Kumar[1]

[1]University of Minnesota, {nayak013, jiaxx221, kumar001}@umn.edu
[2]LinkedIn, vamithal@linkedin.com

Abstract

Time series classification algorithms designed to use local context do not work on landcover classification problems where the instances of the two classes may often exhibit similar feature values due to the large natural variations in other land covers across the year and unrelated phenomena that they undergo. In this paper, we propose to learn discriminative patterns from the entire length of the time series, and use them as predictive features to identify the class of interest. We propose a novel neural network algorithm to learn the key signature of the class of interest as a function of the feature values together with the discriminative pattern made from that signature through the entire time series in a joint framework. We demonstrate the utility of this technique on the landcover classification application of burned area mapping that is of considerable societal importance.

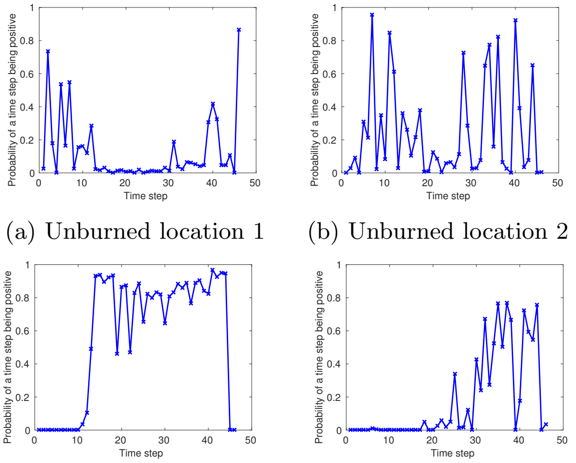
1 Introduction

Multivariate time series classification is an important problem because of the prevalence of temporal data in several domains including signal processing, economics, bioinformatics and remote sensing [1, 2, 3, 4]. The problems in the domain of remote sensing, which serve as the motivation for this work, involve using data collected by earth orbiting satellites to automatically monitor different natural and man-made phenomena across the globe such as deforestation, forest fires, urbanization and crop land mapping. Accurate accounting of land usage helps to ensure that depleting natural resources such as forests and fresh water are judiciously used. Thus, these problems are of great environmental and societal importance.

Landcover classification is the problem of distinguishing a class of interest like burned forests from one or multiple other classes like unburned forests, farms, water etc. One of the key challenges with landcover classification is that using features from a small segment of the year is insufficient. Due to the large variation in the spectral signatures of landcover classes across time and space, assigning the class of the location based on small subsequences of its multivariate time series may be misleading [5]. This is because subsequences of time series belonging to locations of different classes might look similar. However, the time series will most likely not look similar all through the length of time. This is why we propose to model global sequence level patterns instead of capturing local subsequence level context. For example, consider the problem of identifying burned forests. Burning a forest scars the ground and creates a visible difference in its signature in the multivariate sensor data from the satellite. The signature of this burn scar in the sensor data, although very distinct from the signature of healthy forests, can also be exhibited by landcovers like wetlands due to other phenomenon that they undergo and hence can be falsely classified as burned forests by a classifier that just looks at subsequence-level patterns.

For 4 instances (3 unburned, 1 burned) in South CA, figure 1 shows the probability of a scar signature given the features at every 8-day time step in 2008. This probability was obtained from a logistic regression classifier trained using per time step labels (these were hand crafted manually, usually only sequence level labels are available). For the burned location in figure 1d, the probability is close to 0 in the first half of the year but it is high in the second half of the year, which is expected since this is the time of the year when the fires happened in CA in 2008. We would expect the unburned locations to show close to 0 probability all year and most unburned instances do. However, a few like the ones shown here exhibit burn scar like signature on several time steps of the year. Moreover, since burned area is a rare class, even a small number of such instances significantly impact the precision of the classification output.

In this paper, we propose to learn discriminative



(a) Unburned location 1 (b) Unburned location 2

(c) Unburned location 3 (d) Burned location

Figure 1: Probability estimates of an per time step classifier for negative and positive instances corresponding to the problem of burned area classification for the 2008 forest fires in California.

patterns from the entire multivariate time series and use them as predictive features to identify the class of interest. The example in figure 1 shows that just looking at local context such as the fire season time steps will not suffice because of instances like in figure 1c. Thus it is important to look at all time steps. At the same time, the time steps in the fire season are more important than the non-season time steps and hence, should be given more weight when determining the sequence label. This notion of seasonality is key to all landcover classification tasks[1]. In this paper, we propose a neural network algorithm that models a two stage classifier - the first stage learns a transformation on the per time step features, and the second stage learns the weights on each time step based on how discriminative they are in predicting the sequence label. Table 1 shows the discriminative pattern formed by the per time step transformations for the class of interest and a sample of patterns in the negative class for 2 different landcover classification problems. For the burned area classification task, the patterns can be considered to represent the presence of burn scar at each time step. Cropland mapping is another landcover classification task where the patterns can be considered to represent the greenup cycles of the crop being classified. We demonstrate how an explicit modeling of seasonality in a simple model outperforms other time series classification methods in cases where training samples are not available in plenty, which is common in landcover classification tasks. The algorithm proposed in this paper is also a key component of a framework used to build a comprehensive burned area product for the carbon-rich tropical forests.

2 Related Work

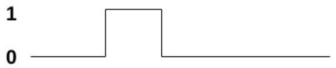
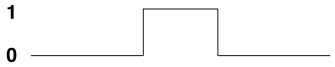
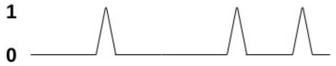
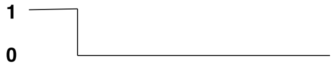
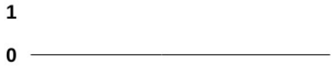
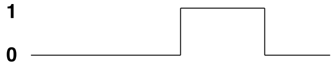
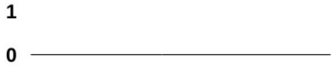
Extensive work has been done in time series classification. The nearest neighbor classifier with a given distance measure defined on sequences is a popular method owing to its non-linear mapping capability and since it does not require extensive parameter tuning [6]. Dynamic time warping is a distance measure defined on sequences that is routinely combined with the nearest neighbor classifier, since it robust to stretching and compression in a sequence [7, 8]. However, these methods may not perform well on multivariate time series especially when the length of a time series gets longer since they rely on reliable computation of distance between sequences.

Ye and Keogh [13] introduced the concept of using subsequences of time series as a primitive for describing each class of sequences. These subsequences are called shapelets and the distance of a time series to a shapelet is used as a predictive feature to assign class membership. Using all contiguous subsequences of time series in the training data set as candidates, the subsequences that rank higher according to the information gain criteria over the target class are chosen as shapelets. Shapelets have been shown to perform really well in a variety of applications with univariate time series. However, the main philosophy behind shapelets is that the discriminating information for a target class lies in local variations and there is not much to be gained in looking at the global context [13, 14, 15]. As we described in section 1, this is not true in many remote sensing problems.

A popular way to model sequence data is through a Hidden Markov Model (HMM). HMMs are generative models where the conditional distribution of the observed features given the latent state of the sequence at each time step is learned. A linear dependence in time among the latent state variable is assumed. HMMs have been used for classification where a different HMM model can be learned for each class and using Bayes rule to combine this with the class priors yields the posterior probability. However, in the presence of limited training data as is routinely the case with remote sensing applications, especially when doing global scale studies, generative models like HMMs may not be accurate enough. A discriminative approach to adapt HMMs to classification problems is through the use of Fisher kernels [9, 10, 11] that are dot products on the transformed feature space of gradients induced by a feature vector on the generative probability function $P(X|\theta)$. These are

[1] Note that these seasons keep changing over space and time and cannot be known beforehand.

Table 1: Sequence level discriminative patterns to identify the class of interest

Class of interest	Burned Area	Corn fields
Signature captured	Burn scar	Greenup of corn
Pattern observed in sequences of positive instances	 (True burn showing scar in summer)	 (Corn field showing peak greenup during mid year)
Examples of patterns observed in sequences of negative instances	 Location with noisy features  (Burned location from last year that still shows scar)  (Unburned vegetation)	 (Soybean fields with their greenup cycle later in the year)  (Unplanted fields)

known to perform at least as good as a MAP decision rule given the same generative model [9].

Conditional Random Field (CRF) is a discriminative approach to modeling sequences that is popular in several domains including natural language processing [16, 17, 18]. They are typically used in cases where the problem is to predict labels for every time step in the sequence given the features at that time step. Such a CRF model is trained using sequences with features and labels available for each time step, as against our case where labels are only available at the sequence level. Some extensions of CRF to sequence classification have been proposed, such as the hidden state CRF (HCRF) [19] where a chain of latent variables models the hidden states and the sequence label is then modeled as a function of these hidden states. Although the intuition of methods like HCRF is similar to our proposed approach, the assumption of a linear dependency among the latent state variables may not necessarily hold in our case as shown in the examples in section 1.

3 Method

3.1 Problem Setting In this work, we propose a method to classify fixed length time series into one of 2 given classes. Consider a data set X consisting of N time series $\{X^1, \cdots, X^T\}$. Every time series is of fixed length T. Let $X_t^i \in R^D$ denote the feature vector corresponding to the tth time step of the time series X^i. We aim to learn a classifier $f : X^i - > Y^i$ that maps a given time series X^i to its class label $Y^i \in 0, 1$. In the context of our application, $Y^i = 1$ corresponds to the class of interest (e.g burned forest) that we are trying to identify and $Y^i = 0$ corresponds to the other landcovers (e.g unburned forests).

3.2 The SeqRep model Figure 2 shows the proposed neural network model. The model has $T \times (D+1)$ nodes in the first layer, corresponding to the D features and a bias term for each time step. The model has a hidden layer with T nodes $\{h_1, \cdots, h_T\}$. Each one of these hidden nodes corresponds to a time step. Unlike a regular neural network with one hidden layer, the hidden layer nodes h_t corresponding to time step t are connected only to input (first layer) nodes for that time step. Thus, there is a correspondence between every hidden layer node and a time step. Moreover, the weights $\{\beta_0, \beta_1, \cdots, \beta_D\}$ connecting the input layer node at a time step to the corresponding hidden layer node are assumed to be shared across time steps. Finally, there is an output node O that is a function of all the per time step predictions captured in the T nodes in the hidden layer. Weights $\{w_0, w_1, \cdots w_T\}$ connect the T hidden layer nodes along with a bias term to the output node O. The hidden layer nodes can be viewed as outputs from a classifier that maps the features at a time step (e.g., reflectance values) to the per time step predictions (e.g., How scarred does the location look on this time step?). The vector of these per time step predictions are then combined to output the prediction for the whole time series (e.g.,does the time series represent a burned location?).

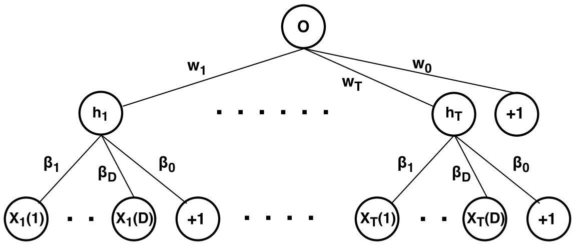


Figure 2: SeqRep model

3.2.1 Activation functions The activation function at the hidden layer is modeled as sigmoid function i.e, the hidden layer node h_t for the tth time step is computed as,

$$h_t = \frac{1}{1 + exp\left(-\left(\sum_{d=1}^{D} \beta_d X_t(d) + \beta_0\right)\right)}$$

where $X_t(d)$ represents the dth feature for the tth time step. Similarly, given the values for hidden layer nodes$\{h_1, \cdots, h_T\}$, the value for the output node O corresponding to the prediction for the whole time series is computed as,

$$O = \frac{1}{1 + exp\left(-\left(\sum_{t=1}^{T} w_t h_t + w_0\right)\right)}$$

3.2.2 Training the model Given a training set of N time series with corresponding features $\{\{X_t^i\}_{t=1}^T\}_{i=1}^N$ and labels $\{Y^i\}_{i=1}^N$, the values for the weights $\{\beta_0, \beta_1, \cdots, \beta_D\}$ and $\{w_0, w_1, \cdots w_T\}$ are optimized so as to maximize cross entropy. Specifically, we choose the weights that maximize the following cost function,

$$(3.1) \qquad \frac{1}{N}\sum_{i=1}^{N} Y^i \log(O^i) + (1 - Y^i)\log(1 - O^i)$$

where O^i is the model prediction corresponding to the ith time series. This cost function computes the average cross entropy of the model output across all time series. This optimization problem can be solved with batch gradient descent using the standard back propagation algorithm. The weights are initialized to a random value and are iteratively updated using the following equations until convergence.

$$\begin{aligned}
&(3.2)\\
&\quad \text{if } d \in \{1 \cdots D\},\\
&\beta_d = \beta_d + \frac{s_1}{N}\sum_{i=1}^{N}\left(O^i(1 - O^i)\sum_{t=1}^{T} w_t h_t^i(1 - h_t^i) X_t^i(d)\right)\\
&\quad \text{if } d = 0,\\
&\beta_d = \beta_d + \frac{s_1}{N}\sum_{i=1}^{N}\left(O^i(1 - O^i)\sum_{t=1}^{T} w_t h_t^i(1 - h_t^i)\right)
\end{aligned}$$

where the output O^i for each sequence and the values for the hidden layer nodes h_t^i are computed using the parameters from the previous iteration. Similarly, weights $\{w_0, w_1, \cdots w_T\}$ are updated as,

$$\begin{aligned}
&(3.3)\\
&\qquad \text{if } t \in \{1 \cdots T\},\\
&\qquad w_t = w_t + \frac{s_2}{N}\sum_{i=1}^{N}\left(O^i(1 - O^i)h_t^i\right), \qquad t \in \{1 \cdots T\}\\
&\qquad \text{if } t = 0,\\
&\qquad w_t = w_t + \frac{s_2}{N}\sum_{i=1}^{N}\left(O^i(1 - O^i)\right)
\end{aligned}$$

Parameters s_1 and s_2 in the above equations represent step size parameters for the gradient descent algorithm.

3.3 SeqRep model - the heterogeneous variant

In the above model, we assume that the weights that connect the input layer to the hidden layer (representing the per time step classifier) are the same for all time steps. However, this may not necessarily be the case and irrespective of the key signature (e.g, burn scar) being present or not, the features can have some natural variation across time steps and hence, one might want to model the weights at each time step differently. Thus instead of having one set of weights $\{\beta_0, \beta_1, \cdots, \beta_D\}$ from the input layer, we have a different set of weights $\{\beta_0^t, \beta_1^t, \cdots, \beta_D^t\}$ for each time step. We call this variant that models the temporal heterogeneity as SeqRepHet. We still optimize function 3.1 and similar update equations as equation 3.2 and 3.3 can be derived for this case. Note that this is still different from the traditional neural network with one hidden layer in that each hidden layer nodes is still associated with a unique time step.

4 Evaluation

Forest fires are known to generate a significant flux of greenhouse gases and particulate matter into the at-

Table 2: Summary of data sets used in this paper.

Dataset	Land cover of interest	Instances	Skew = $\frac{\#\text{Negatives}}{\#\text{Positives}}$
South CA (2008)	Burned Forests	141,900	56.69
Montana (2007)	Burned Forests	276,310	44.87
South CA (2007)	Burned Savannas	198,287	60.73

mosphere and also contribute to several ecological effects such as the loss of animal habitat and biodiversity [26]. To evaluate the performance of our algorithm, we used it perform three classification tasks, each identifying a different class of interest namely 1) Burned forests in South California 2) Burned forests in Montana 3) Burned Savannas in South California. One of the reasons for choosing US for ground truth based validation is the availability of good quality validation data in this region. Government agencies like Cal Fire are responsible for monitoring and managing forests and wildfires [22]. The validation data is in the form of fire polygons, each of which is associated with the time of burning. We consider an event to be positive if the corresponding pixel lies completely inside a polygon. Similarly, an event is considered to be unburned (forming the negative class) only if the entire pixel is outside a polygon. Since it is difficult to decide the class (burned/unburned) for a pixel which is partially inside the polygons, pixels that partially overlap polygon boundaries are discarded from the evaluation framework to avoid ambiguity. The details of these data sets is given in table 2.

For each one of these problems, every 0.25 sq.km area on the ground is an instance we are trying to assign a landcover class label to. We use the time series of reflectance data collected over a year at each location as features. Specifically, we use 500m-resolution MODIS data product that captures the reflectance at every location in 7 bands (620-2155 nm) collected by sensors aboard the Terra and Aqua satellites [20], thus every location has a multivariate observation at every time step. We use the level-2 reflectance product that has been georeferenced and corrected for sensor induced errors. This product provides images at a global scale for every 8 days. All datasets used in this paper and codes for SeqRep can be downloaded here.

4.1 Model complexity The SeqRep model that has $(D + T + 2)$ parameters. The SeqRepHet model subsumes this model and has $(DT + 2T + 1)$ parameters. The conventional neural network with one hidden layer of $DT^2 + T + 2$ nodes further subsumes SeqRepHet and has T parameters. As the number of parameters increases, the space of decision boundaries modeled increases. However, the number of samples needed to train the model to reach its optimum performance also increases. Figure 3 shows the variation in the performance of these three schemes as the number of training samples are varied on one of the test cases. It can be seen that when the number of training samples is low, SeqRep has better F-measure and lower standard deviation. SeqRep also reaches it's optimum performance in fewer samples. However, given a large number of training samples, SeqRepHet starts to approach the performance of SeqRep. Since, SeqRepHet and ANN are more complex models, given enough training samples, these schemes will eventually perform at least as good as SeqRep that assumes a certain structure to hold true in the data. Note that, it may not always be possible to find a large number of training samples for every landcover around all geographies around the globe. As we see, the ANN performance is still increasing slowly but we did not have enough training samples for this case to see it through the end.

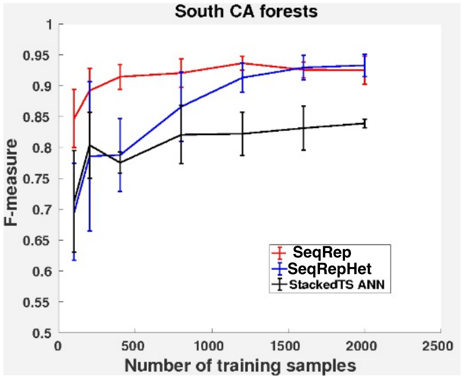


Figure 3: Figure shows the variation in F-measure (mean and standard deviation over 5 runs) with respect to the number of training samples for the 3 schemes for mapping burned forests in South California

4.2 Comparison with baselines

4.2.1 Baselines In our experiments, we compare the performance of our method to the following baselines -

1. **1 NN + DTW:** The nearest neighbor classifier with dynamic time warping distance is a widely used classifier for time series data.

2. **Hidden CRF:** The Hidden Conditional Random Field (HCRF) [19] is a discriminative classifier that can be used to model the sequential nature of time series data [19], [23]. We use the implementation provided by the authors in [19].

3. **FKL:** Fisher Kernel Learning (FKL) [11] uses the fisher kernel based on gradient on the parameters of a hidden markov model (HMM) learned on the time series. It has been shown to perform well on a range of time series classification problems. We use the SVM classifier on top of the kernel. We use the implementation provided by the authors in [11].

4. **Shapelets:** Shapelets capture the notion of using local context to classify time series and have been shown to be really effective for several, mostly univariate, time series classification tasks. We compare with the shapelet discovery algorithm proposed in [14] since it is shown to work well on multivariate time series classification tasks. We use the implementation provided by the authors in [14].

5. **Stacked TS ANN:** This is a neural network with one hidden layer. This is similar to our model except that our model assumes that the hidden layer nodes are only connected to the corresponding input nodes and the weights of these connections are the same across all time steps. We include this baseline because in principle, if provided with a large number of training samples, this baseline can capture the structure that we hard coded in our design.

6. **Long-short Term Memory (LSTM)**: Here we implement recurrent neural networks with LSTM cell using Keras library. This baseline method is capable of capturing long-term temporal patterns over long multi-variate sequence by introducing an internal memory flow and has shown tremendous success in various applications, including natural language processing, health-care records, etc [27]. However, this method involves large number of parameters and is vulnerable to overfitting.

In reporting our results, for methods that output a continuous score instead of a binary output, the threshold that maximizes the desired metric (F1-score) on the test set is used as a threshold to binarize the output and compute the evaluation metric.

4.2.2 Results In each experiment, we use 500 locations (250 burned, 250 unburned) for training the classifier. The remaining locations were used for testing. Since, burned area is a rare class, we use F1-score as the evaluation metric. Table 3 reports the average F1-score corresponding to 5 random partitions of the data set into train, and test sets for each of the 3 problems. Our method clearly outperforms the baselines for these tasks. Because of the rarity of the burned class, even a small number of false positives can result in a low precision and hence a low F1 score. Hence, it is important for the classifier to be really precise. For example, the accuracy of Shapelets baseline on the problem of detecting burned forests in South CA on the test set is 0.94, and its false positive rate is only 0.08, but since the skew of this data set as shown in table 2 is 56.69, the resulting F1 score is poor. Moreover, methods like HCRF and LSTM, albeit powerful, need a lot of samples to attain optimal performance and do not perform well in limited training sample scenarios, which is the case in landcover classification problems. Also, inverse to the order of model complexity, SeqRep has a slightly better performance than SeqRepHet which is significantly better than the stacked ANN baseline.

Table 3: Table shows the mean of F-measures for each algorithm for different test cases computed over 5 random partitions of training and test sets for each of the 3 burned area mapping tasks.

Algorithm	South CA Forests	Montana Forests	South CA Savannas
StackedTS ANN	0.72	0.78	0.77
DTW + 1NN	0.42	0.27	0.52
Shapelets	0.34	0.32	0.66
HCRF	0.19	0.17	0.46
FKL + SVM	0.27	0.26	0.38
LSTM RNN	0.44	0.50	0.33
SeqRepHet	**0.88**	**0.86**	**0.91**
SeqRep	**0.91**	**0.93**	**0.93**

4.3 Learning weights on time steps Methods SeqRep and SeqRepHet capture the classifier at every time step $\{\beta_0, \beta_1, \cdots \beta_T\}$ jointly with the weight w_t assigned to each time step t based on its discriminating capacity. In this section, we experimentally justify our logic of giving different weights to all time steps and learning these weights in a joint framework with the per time step classifiers. One way to judge the discrimination of each time step is to learn a classifier $f_t : X_t - > Y$ that maps the features at that time step to the sequence label. The accuracy on the training set for each of these T classifiers can be used as a proxy for how discriminating each time step is. We illustrate this for the test case of detecting burned forests in South CA. We learned a different logistic regression classifier on every time step using a balanced training set of 500 samples. Figure 4 shows the plot of the error on the training set for the classifier at each time step. Using this idea of per time step classifiers, we consider 3 classifiers that lead up to our scheme -

Method 1 - Use the best time step One way to classify the sequence would be to consider the time step

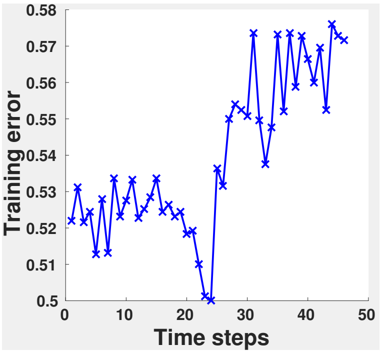


Figure 4: Accuracy for classifiers trained to predict the time series label using features at a single time step.

with the maximum accuracy. From figure 4, time step 45 is the one with the maximum accuracy. So, we could use the classifier trained on features from time step 45 to predict the time series label.

Method 2 - Use all time steps, treating them equally In this method, we generate predictions from classifiers at every time step and then use max aggregation to generate time series prediction. So, if the classifier at any time step calls the time series as positive, we assign it a positive class label. This is better than method 1 because it is taking all time steps in to account.

Method 3 - Use the predictions from each time step as predictive features In this method, we use the idea of SeqRepHet where we have a classifier at each time step and another logistic regression classifier is learned over the vector of predictions from each time step. This is better than method 2 because in addition to considering multiple time steps, it is also weighting them appropriately.

We evaluate these methods on the test set consisting of all instances not used for training. The F1 score for method 1 is 0.31 which shows that it is not sufficient to just consider the most discriminating time step. Method 2 that considers all time steps, but treats them equally has a F1-score of 0.39. Figure 5 shows the performance of the method 3. Here, we order the time steps according to their discriminating they are according to the per time step classifier. For the top k most discriminating time steps, we take the vector of predictions from their respective per-time step classifiers and learn a second classifier that maps this vector to the time series label. Figure 5 shows the variation in F1-score on the test set as the number of time steps k is increased. F1 scores for other schemes including method 1, method 2, SeqRep and SeqRepHet are added for reference. The plot starts from where method 1 left us and considering the top few (5) time steps, that correspond to the fire season, takes the F1-score from 0.31 to about 0.7. However, it is interesting to see that the 23rd time step in the ordering (actual time step 14) contributes significantly to the F1-score (a jump of almost 0.1) even though its per time step classifier is not discriminating enough. The F1-score when we consider all time steps is 0.81. However, this is still lower than the performance of SeqRepHet (0.88) on the same data set. Note that SeqRepHet is doing the same thing as this method, except that the per time step classifiers and the weights on each time step are learned in a joint fashion. This is the reason for the difference in their F-scores.

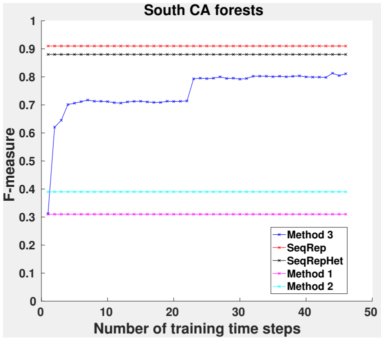
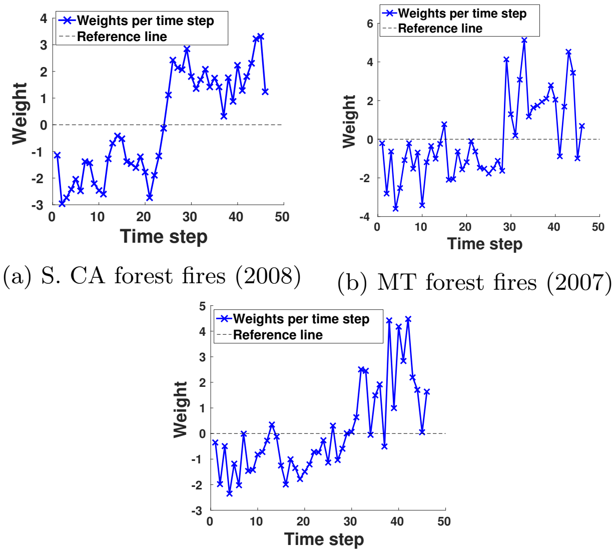


Figure 5: Variation in F1-score as features from more time steps are used for classification using method 3 in section 4.3. The method learns weights w and per time step classifier β independently and hence has suboptimal performance as compared to SeqRepHet. F1 scores for other schemes including method 1, method 2, SeqRep and SeqRepHet are added for reference.

4.4 Interpreting weights on time steps Figure 6 shows the per time step weights i.e the vector w learned for the 3 tasks of mapping burned area. The vector w captures the weight placed on each time step when the pattern of per time step scores from the whole time series is mapped onto the final time series prediction. We expect that discriminative time steps will be assigned higher magnitude weights while the weights assigned to non-discriminative time steps should be close to zero. Figure 6a shows the plot of the weights connecting the hidden layer to the output node corresponding to each of the 46 time steps in the network learned for classifying the 2008 forest fires of southern California. As expected, time steps 23-39 in the second half of the year (time steps 23-46) have been assigned higher positive weights since these correspond to the forest fire season in this region. Some time steps get assigned close to zero weight because they're probably not very discriminating. However, it is interesting to note that some time steps in the first half get assigned high magnitude negative weights. This implies that if a location looks burned at this time step,



(a) S. CA forest fires (2008)

(b) MT forest fires (2007)

(c) S. CA savanna fires (2007)

Figure 6: Figure showing the weights connecting the hidden layer to the output node in the network learned for the 3 tasks of identifying burned area.

the model reduces the probability of the location being burned.

Similar weights are learned for the 2007 savanna fires as shown in figure 6c. The probable reason for this is the fact that the state of California experiences fires each year and some of these fires are strong enough that the scars from them do not recover for multiple years. So, several locations that did not experience any burn activity this year might look burned in the first half of the year because of the fires they experienced last year. However, our model has learned from the training set that the first half of the year is not the true fire season in this region.

Figure 6b shows the weights learned for the problem of mapping forest fires in Montana. This case also shows high weights in the beginning of the second half of the year when the fires occur here. However, unlike the other two cases, the first half does not have a consistent stretch of negative weights of high magnitude. This is probably because the primary cause of confusion in this region is not previous year fires, but the snow that covers the ground in winter. Note that snow is not an exact match to burned ground (and it does not always cover the ground, hence we do not see consistently strong negative weights) but in the spectral space, it is much more similar to burned ground than to healthy forests.

5 SeqRep at work

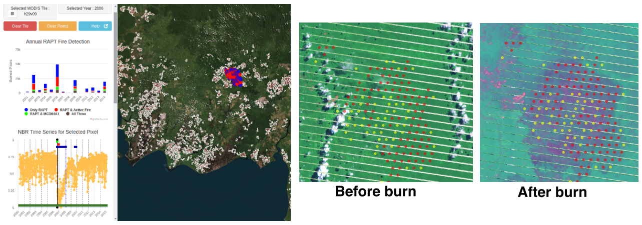
In section 4, we demonstrated the performance of the proposed method on test regions in the US. The primary reason for this is that the US government has the resources to invest in developing these landcover maps via agencies like CalFire [22] that deploy huge manpower and resources to create these maps using systems that are not fully automated. Our goal in developing methods like the one proposed here is to develop landcover maps on a global scale in an automated fashion.

This method is part of a two-component framework that has been used to build a comprehensive product [25] for the carbon rich tropical forests. Tropical forest fires contribute significantly to carbon loss and are often associated with active deforestation fronts and linked to illegal establishment of industrial timber, oil palm, soy, and tea and coffee plantations [21]. Mapping burned area in the tropics is challenging because of the number of confounding factors like the poor quality of landcover maps, occlusion by smoke and clouds is much more in the tropical areas. The other factor is that the available training labels are very noisy. In [24], we proposed a method to handle the noisy nature of training labels. That work is used in conjunction with the classifier proposed in this paper to develop a product which has three times the coverage of the burned area reported by the state-of-the-art NASA product MCD64A1 [26] while having comparable or better precision.

In [25], we provide extensive validation of the burned area product via semi-automated inspection of high resolution imagery of randomly sampled smaller areas in the region. In addition, we have made our burned area maps accessible to ecologists and to the general public via a web viewer http://z.umn.edu/fireviewer. The web interface lets you inspect every 0.25 sq.km pixel in our burned area maps for the high carbon tropical forests of southeast Asia and Amazon for every year since the beginning of MODIS program in 2001. For each location, users can judge the quality of the detection by examining its temporal profile via time series of indices like Enhanced Vegatation Index (EVI), Normalized Burned Ratio (NBR) along with the best high resolution imagery available for that region around the date of the fire. We believe that such comprehensive publicly available burned area maps are critical in ensuring that we deploy sustainable means to cultivate plantations in the region, which is crucial for protecting these forests and the wildlife that depends on it.

6 Conclusion

In this work we presented a multivariate time series classification algorithm that learns discriminative patterns from the whole length of the time series rather than looking at local contexts. We demonstrate the utility of the application on real world problems of mapping burned area in the US where validation data is read-



Selected MODIS Tile : h29v09

Selected Year : 2006

Figure 7: Figure shows the screenshot of the web viewer (http://z.umn.edu/fireviewer) for the burned area product in the tropics and the manual validation using high resolution (30 m) Landsat images that it enables to handle lack of validation data.

ily available. The method is also a key component of the framework used in developing a burned area product [25], which is much more comprehensive than the state-of-the-art, in the more challenging and important region of tropics. In the future, this method can be extended to work with variable length time series and variable temporal frequency of features. This can happen because of multiple data sources with different time span or frequencies of data acquisition.

Acknowledgement This research was supported by National Science Foundation under Grants IIS-1029711 and NASA grant NNX12AP37G. Access to computing facilities was provided by the University of Minnesota Supercomputing Institute and NASA Earth Exchange (NEX).

References

[1] Caiado. J, N. Crato, and D. Peña. "A periodogram-based metric for time series classification." Computational Statistics & Data Analysis (2006)

[2] Lin, Tien-ho, N. Kaminski, and Z. Bar-Joseph. "Alignment and classification of time series gene expression in clinical studies." Bioinformatics 24.13 (2008)

[3] Povinelli, Richard J., et al. "Time series classification using Gaussian mixture models of reconstructed phase spaces." IEEE TKDE 16.6 (2004)

[4] Karpatne, Anuj, and Stefan Liess. "A Guide to Earth Science Data: Summary and Research Challenges." Computing in Science & Engineering 17.6 (2015)

[5] Chandola, V., and R.R. Vatsavai. "Multi-temporal remote sensing image classification-A multi-view approach." CIDU (2010)

[6] Ding, Hui, et al. "Querying and mining of time series data: experimental comparison of representations and distance measures." VLDB (2008)

[7] Itakura, Fumitada. "Minimum prediction residual principle applied to speech recognition." IEEE Transactions on Acoustics, Speech, Signal Processing (1975)

[8] Niennattrakul, Vit, and C. Ann Ratanamahatana. "On clustering multimedia time series data using k-means and dynamic time warping." ICMU (2007)

[9] Jaakkola, Tommi, and D. Haussler. "Exploiting generative models in discriminative classifiers." NIPS (1999)

[10] Jaakkola, Tommi, M. Diekhans, and D. Haussler. "A discriminative framework for detecting remote protein homologies." Journal of computational biology (2000)

[11] Maaten, L. "Learning discriminative fisher kernels." ICML(2011)

[12] Quattoni, Ariadna, et al. "Hidden conditional random fields." IEEE transactions on pattern analysis and machine intelligence (2007).

[13] Ye, Lexiang, and E. Keogh. "Time series shapelets: a new primitive for data mining." KDD (2009).

[14] Cetin, Mustafa S., Abdullah Mueen, and Vince D. Calhoun. "Shapelet ensemble for multi-dimensional time series." SDM (2015).

[15] Grabocka, Josif, et al. "Learning time-series shapelets." KDD (2014).

[16] Lafferty, John, A. McCallum, and F. CN Pereira. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data." (2001).

[17] Morency, Louis-Philippe, A. Quattoni, T. Darrell. "Latent-dynamic discriminative models for continuous gesture recognition." CVPR (2007)

[18] Artieres, Thierry. "Neural conditional random fields." AISTATS (2010)

[19] Quattoni, Ariadna, et al. "Hidden conditional random fields." IEEE transactions on pattern analysis and machine intelligence (2007).

[20] The MODIS data product was retrieved from the online Data Pool, courtesy of the NASA Land Processes Distributed Active Archive Center (LP DAAC), USGS/EROS Center.

[21] Fuller, D. O., and M. Fulk. "Burned area in Kalimantan, Indonesia mapped with NOAA-AVHRR and Landsat TM imagery." IJRS (2001).

[22] Eidenshink, J., et al. "A project for monitoring trends in burn severity. Fire Ecology 3 (1): 3-21." Fire Ecology Special Issue Vol 3 (2007).

[23] Kim, Minyoung. "Semi-supervised learning of hidden conditional random fields for time-series classification." Neurocomputing (2013).

[24] Mithal, V., Nayak, G., Khandelwal, A., Kumar, V., Oza, N. C., & Nemani, R. (2017). Rapt: Rare class prediction in absence of true labels. IEEE Transactions on Knowledge and Data Engineering, 29(11).

[25] Mithal, V.; Nayak, G.; Khandelwal, A.; Kumar, V.; Nemani, R.; Oza, N.C. Mapping Burned Areas in Tropical Forests Using a Novel Machine Learning Framework. Remote Sens. 2018, 10, 69.

[26] Giglio, Louis, et al. "An active-fire based burned area mapping algorithm for the MODIS sensor." Remote Sensing of Environment (2009)

[27] Sutskever, Ilya, O. Vinyals, and Q.V. Le. "Sequence to sequence learning with neural networks." NIPS (2014).