TEMPORAL MAPPING OF HYPERSPECTRAL DATA

Ronald Fick[A], Paul Gader[B], Alina Zare[C], Susan Meerdink[D]*

University of Florida[A,B,C,D]
Department of Computer and Information Science and Engineering[A,B,D]
Department of Electrical and Computer Engineering[C,D]
Engineering School of Sustainable Infrastructure and Environment[B,D]

1. ABSTRACT

The increasing popularity of hyperspectral sensors is dramatically increasing the temporal availability of data. To date, algorithms struggle to compare hyperspectral data collected across dates due to different environmental conditions during collection. In this work, we develop a temporal mapping in order to map data collected from one year to a different year. We investigated both conditional generative adversarial networks (cGANs) as well as affine transformations to perform this mapping. Both methods showed an improvement over using data from past collections without mapping, with cGANs outperforming the affine transformation.

2. INTRODUCTION

Hyperspectral data is becoming increasingly available in time-series in recent years. In order to fully leverage this existing dataset and decrease future processing time, there has been an increased desire to use the available data collected from previous years when processing data collected more recently. In this work, we investigate attempting to map data from previous years in order to use it to with new data. Directly applying data from previous collections to new collections will almost certainly prove unsuccessful. This occurs because data collected at different times will be under different environmental effects, different illumination conditions, as well as material changes. Section 5 will show this effect effect empirically.

Ideally, we could produce a mapping which accounts for these temporal effects and allows past data to be compared directly with new data. This work investigates producing such a mapping. The paper is organized as follows. Section 3 will discuss the details of the methods used, section 4 will outline the experiments performed, and section 5 will show the results of the experiments performed.

*The fourth author performed work while at the University of California, Santa Barbara

3. TECHNICAL METHODS

3.1. Generative Adversarial Networks

Generative Adversarial Networks (GANs) are networks introducted by Goodfellow et al. in 2014 [1] that learn to generate data given samples from a distribution. These networks are actually composed internally of two separate neural networks, one is referred to as the generator, and the other as the discriminator. The discriminator is trained to differentiate between samples taken from the data and samples generated by the generator. The generator is trained to fool the discriminator by increasing its error rate.

GANs have been successfully applied in a wide variety of image generation tasks. For example, GANs have been applied to do image superresolution [2], create images from an input string of words describing those images [3], and convert magnetic resonance images (MRI) to computed tomography (CT) images [4].

The objective function used to derive the training algorithm has two terms. The first term is designed to be optimal when the Discriminator, D, produces large output values when true training samples, $\mathbf{x} \sim p_X$, are used as inputs. The second term has two competing, or *adversarial* objectives. Maximizing with respect to Θ_D encourages the outputs of D to be small when outputs from G, the Generator, are used as inputs to D. Minimizing with respect to Θ_G encourages the outputs of D to be large when outputs from G are used as inputs to D. The full objective function is given in equation 1.

$$\min_{\theta_G \in \Theta_G} \max_{\theta_D \in \Theta_D} J(\theta_D, \theta_G) = \mathbb{E}_{\mathbf{x} \sim p_X(\mathbf{x})}[logD(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_z(z)}[log(1 - D(G(\mathbf{z})))] \quad (1)$$

3.2. Conditional Adversarial Networks

The Conditional Generative Adversarial Network (cGAN) is a modification to the standard GAN proposed by [1]. The goal of cGANs is to add additional information into the system so that the generation process can be guided to a desired result.

An example of this could be to give the network a wireframe image and have the network fill in the wireframe, as was done in [5].

cGANs are trained to optimize the objective in equation 2. In this equation, u refers to an observed vector used to guide the data generation. If we compare this to the original objective function from [1] in equation 1, we can see that the difference is in the mapping of the generator. In the original GAN formulation, the generator G is learning a mapping $G : Z \times \Theta_G \rightarrow \mathbb{R}^B$ from the space of random vectors Z to the data space $\mathbb{R}^B$. In the cGAN objective, the generator's mapping becomes $G : (u, Z) \times \Theta_G \rightarrow \mathbb{R}^B$.

$$\min_{\theta_G \in \Theta_G} \max_{\theta_D \in \Theta_D} J(\theta_D, \theta_G) = \mathbb{E}_{\mathbf{x} \sim p_X(\mathbf{x})}[logD(u), \mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_z(z)}[log(1 - D(u, G(u, \mathbf{z})))] \quad (2)$$

Note that the vector u can arise from a variety of data sources. In [5], u was an image, while [6] used u as a text description of an image, and [7] used class labels for u.

3.3. Affine Transformation

The method we are using to compare with the cGAN is affine transformations. We apply a transformation of the form in equation 3.

$$\overrightarrow{y} = \mathbf{A}\overrightarrow{x} + \overrightarrow{b} \quad (3)$$

The parameters $\mathbf{A}$ and $\overrightarrow{b}$ are learned using least squares regression over all available training data. To be clear, we take all data from a previous data collection and from a more recent data collection for which the same polygons can be found in both datasets. The data from the past collection forms $\overrightarrow{x}$, and the data from the more recent collection forms $\overrightarrow{y}$.

4. EXPERIMENTAL DESIGN

The goal of this experiment is to characterize the performance of the models at performing temporal mapping of hyperspectral data. We perform experiments on data collected during the Hyperspectral Infrared Imager (HyspIRI) Airborne Preparatory Campaign. Details of the HyspIRI dataset are provided in [8]. We summarize relevant details of that discussion here. These images were collected using the Airborne Visible / Infrared Imaging Spectrometer (AVIRIS) sensor, which measures reflectance between 360 and 2500 nm [9]. The data has a spatial resolution of 18 m.

The Santa Barbara flightbox, which is the focus of this study, is pictured in figure 1. These flightlines cover a range environments, from coastal Santa Barbara to the Los Padres National Forest. This data consists of a time series which was

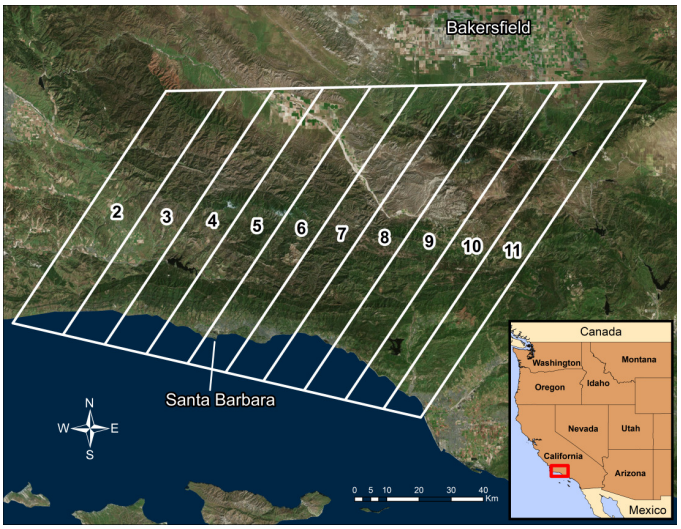

Fig. 1. AVIRIS HyspIRI Flight Lines (image from [8]).

collected three times per year in April, June, and November or August during 2013, 2014, and 2015.

The aspect of the HyspIRI data which makes it especially suitable for this work is the reference data which was collected by [8]. This reference data was collected using a rule that patches must contain at least 75% of a given species to be labeled as that species. The reference data consists of polygons covering more than 5 pixels per polygon. [8] collected 700 reference polygons to go along with 400 polygons collected from previous studies.

In this study, we focus on data from April 11^{th} 2013, April 16^{th} 2014, June 6^{th} 2013, and June 6^{th} 2014. The goal is to map the April 2013 data to April 2014, and June 2013 to June 2014. In order to assess the mappings quantitatively, we will test them with a classification task.

All classification will be done using a K-nearest neighbor (KNN) classifier. Two baseline models will be trained. The first will be using both training and test data from the 2014 data. The second will be using training data from 2013 and test data from 2014. The proposed method's performance will be tested by mapping training data from 2013 to 2014, and using that mapped data to classify data from 2014. The data has 27 classes, corresponding to various plant species. Classification results will be assessed in terms of correct classification rate, as well as top 5 classification rate.

Data from both dates will be split into training and testing sets, with 80% training and 20% testing. This split of training and testing was stratified by class and polygon. To clarify, all pixels from a single polygon was present in either the training or testing set, there were never pixels from the same polygon in both sets. Additionally, the split was done on a per class basis to ensure that each class has sufficient polygons in both sets.

In order to train the cGAN model, we need to supply it

with paired data points from domain A and domain B for training. Across different dates, we have access to polygons corresponding to the same individual plant species. However, due to residual geolocation error, we have chosen to pair pixels from the same polygon across two different dates at random.

5. EXPERIMENTAL RESULTS

We utilize dynamic time warping (DTW) as one method of evaluation. DTW attempts to optimally match two different sequences under several criteria, namely that the first and last indicies of both sequences must be paired, and indices from both sequences must be mapped monotonically increasing. The matching produced by DTW can give us a distance between two spectra to understand their similarity. Smaller distances show that the spectra are more similar.

The goal is to understand how well the algorithms' output distribution after being given data from 2013 matches the real data from 2014. We sample 20 points at random from each class for each approach, as well as 20 points from the real data and compute the average of all pairwise distances according to DTW. Table 1 summarizes these results. The approaches which improve on the baseline for a given class are shown in blue, those that get worse are shown in red. The bottom row of the table shows the average improvement over the unmapped data, or how much smaller the DTW distance is for that method than for the unmapped data.

Tables 2 and 3 summarize the classification results of this study. Each table refers to a different time of year that was analyzed. The first row of each table refers to the result obtained if we train and test on data from the same date. This will always give the best performance, the goal is to see how close the other approaches can get to this performance.

The rows labeled 'Baseline' refer to training on the raw data from 2013, and testing with data from 2014 with no modifications. The rows labeled 'Affine Transformation' rows refer to passing the data from 2013 through a affine transformation of the form 3, training on the converted data, and testing with the 2014 data. Finally, the rows labeled 'cGAN' refer to passing the data from 2013 through a cGAN, training on that converted data, and testing with the 2014 data.

6. DISCUSSION

Both of the methods investigated outperform the baseline approach in terms of classification results, as shown in tables 2 and 3. Additionally, both methods outperform the baseline according to average DTW distance from table 1. The cGAN approach shows an improvement over affine transformation both in terms of classification error as well as average DTW improvement. The cGAN is using a more complex mapping than the affine transformation, so it makes sense that it would have improved performance. Note that the cGAN does require significantly more computational time than the affine transformation, so that may be a concern depending on the application.

It should be mentioned that the time period of this study was during a drought in California. 2014 was the peak of the drought, which could impact the spectra from that year enough that the signatures are no longer comparable from year to year. It would be interesting to attempt this mapping under different conditions to see what if any difference could be observed.

Class Name	Unmapped	AT*	cGAN
Chamise	7.05	6.30	6.88
Agricultural Residue	2.59	2.25	3.59
California Sagebrush & Purple Sage	3.92	2.73	2.43
Manzanita	3.95	3.86	3.01
Four-wing Saltbush & Rubber Rabbitbrush	10.14	9.30	9.22
Coyote Brush	4.96	3.13	3.29
Black Mustard	6.46	4.83	5.21
Buck-brush Ceanothus	4.85	3.27	2.08
Big-pod Ceanothus	11.57	9.53	9.76
Green-bark Ceanothus	3.52	2.25	1.93
Citrus	8.46	6.47	8.44
Calfornia Buckwheat	6.46	3.71	2.33
Eucalyptus	6.18	4.50	5.02
Irrigated Grasses	4.93	5.12	5.26
Juniper	3.49	3.50	3.04
Mediterranean Annual Grasses and Forbs	10.80	8.93	10.99
Avocado	25.89	21.86	13.48
Jeffrey Pine	5.98	5.96	5.38
Single-leaf Pinyon	1.98	3.43	2.02
Gray Pine	5.92	4.55	1.71
Douglas Fir	5.69	4.44	4.17
Coast Live Oak	6.16	4.90	3.40
California Scrub Oak	3.68	3.31	3.36
Valley Oak	6.70	4.97	4.56
Rock	9.64	9.52	10.09
Soil	0.87	1.22	0.72
California Bay Laurel	3.99	6.15	5.24
Average Improvement over Unmapped	N/A	0.96	1.45

Table 1. Average distances from DTW between generated distributions and real data. The column 'Unmapped' refers to comparing the data from 2013 with data from 2014 with no mapping done. Smaller distances indicate a better mapping. *(Affine Transformation)

Training Set	Test Set	Top 1 Error	Top 5 Error
April 16^{th} 2014	April 16^{th} 2014	55.68%	26.15%
April 11^{th} 2013 (Baseline)	April 16^{th} 2014	76.46%	41.57%
April 11^{th} 2013 (Affine Transformation)	April 16^{th} 2014	68.99%	34.60%
April 11^{th} 2013 (CGAN)	April 16^{th} 2014	66.80%	33.61%

Table 2. Results from Spring 2013-2014.

7. REFERENCES

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville,

Training Set	Test Set	Top 1 Error	Top 5 Error
June 6^{th} 2014	June 6^{th} 2014	53.98%	29.79%
June 6^{th} 2013 (Baseline)	June 6^{th} 2014	72.05%	41.56%
June 6^{th} 2013 (Affine Transformation)	June 6^{th} 2014	63.19%	35.20%
June 6^{th} 2013 (CGAN)	June 6^{th} 2014	61.67%	36.02%

Table 3. Results from Summer 2013-2014.

and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[2] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al., "Photo-realistic single image super-resolution using a generative adversarial network," *arXiv preprint*, 2017.

[3] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5907–5915.

[4] Dong Nie, Roger Trullo, Jun Lian, Caroline Petitjean, Su Ruan, Qian Wang, and Dinggang Shen, "Medical image synthesis with context-aware generative adversarial networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2017, pp. 417–425.

[5] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros, "Image-to-image translation with conditional adversarial networks," *arXiv preprint arXiv:1611.07004*, 2016.

[6] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee, "Generative adversarial text to image synthesis," *arXiv preprint arXiv:1605.05396*, 2016.

[7] Mehdi Mirza and Simon Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[8] Susan Kay Meerdink, *Remote Sensing of Plant Species Using Airborne Hyperspectral Visible-Shortwave Infrared and Thermal Infrared Imagery*, Ph.D. thesis, University of California, Santa Barbara, 2018.

[9] Robert O Green, Michael L Eastwood, Charles M Sarture, Thomas G Chrien, Mikael Aronsson, Bruce J Chippendale, Jessica A Faust, Betina E Pavri, Christopher J Chovit, Manuel Solis, et al., "Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (aviris)," *Remote sensing of environment*, vol. 65, no. 3, pp. 227–248, 1998.