North and South American Cloud Regimes from the Geostationary Operational Environmental Satellite (GOES-16)

Nazia Farhat, Katherine Haynes, Jason Stock, Joe Strout

Motivation

The Geostationary Operational Environmental Satellite[1] (GOES-16) is one of an over 40-year history of continuous satellite imagery and data on atmospheric conditions, with continuous high-resolution spatial coverage over American continents. GOES data is currently being used in a variety of weather and aviation forecasting applications; however, due to computational constraints, applying machine learning techniques to geostationary satellite imagery has primarily focused on either varying cloud cover over specific regions (e.g. Andersen & Cermak, 2018) or cloud cover at any vertical level (e.g. Qin *et al.*, 2019; Shang *et al.*, 2018; Wind *et al.*, 2010). Accurately identifying and predicting cloud regimes over large-scale regions, such as North and South America, is an important yet difficult problem that is only now able to be investigated using big data techniques. In this study, we aim to investigate different cloud regimes seen by GOES using deep learning.

Identifying and predicting cloud regimes using GOES data has a wide range of uses for evaluation of climate change and weather process understanding. For example, knowledge of cloud regimes can help identify patterns of rainfall and their sensitivity to climate change. Understanding how cloud structures vary regionally would further our knowledge on the climate system and help predict localized responses to change. Determining cloud regime properties directly from radiances could be used to replace or supplement the processing of radiances into cloud properties. Finally, accurate cloud identification would provide more accurate sky conditions to improve weather forecasts.

Data Overview

For this study, we will use GOES-16 satellite data from 16 different channels (7 visible and 9 infrared). Supplementing the 16 radiance channels, we will use three additional features: cloud top height, optical depth, and effective radius. All three are from the NOAA Enterprise Algorithms processed at the Cooperative Institute for Research in the Atmosphere (CIRA).

GOES covers North and South America along with the surrounding oceans, with a latitude range from 60 S to 60 N. Each pixel consists of 2 x 2 km pixels at the equator that are updated every 10 minutes. To avoid seasonal impacts, we will focus on October 27-November 30, 2017 (6019 files) for a total data size of 2.1 TB. From this dataset, we will subset daytime retrievals only (9 AM to 3 PM LST) to avoid solar zenith angle impacts. To begin our analysis, we will start with a small oceanic subset off the East coast, from 35-41 N and 60-69 W. We plan to use this dataset as a proof of concept for our project, and hope time permits us to then utilize the complete GOES spatial coverage.

The data is currently located on Google Drive. For this project, we plan to download a subset of the data into a HDFS cluster. Since the data is in netcdf format, we plan to follow the approach by Biookaghazadeh *et al*. (2015) to directly utilize netcdf files in both Spark and PyTorch. However, if necessary we will convert the files to comma-separated values for storage and processing.

[1] https://www.goes-r.gov/mission/history.html

Algorithms and Framework

We will perform three unsupervised learning techniques on the GOES dataset:

1) *k*-means Directly
2) *k*-means + Autoencoder
3) Deep Embedded Clustering Algorithm

k-means Directly

To identify different cloud regimes, we will use Apache Spark to perform *k*-means clustering on the three cloud features (top height, optical depth, and effective radius). This analysis will use Spark's ML library and follow their clustering approach[2]. We will originally include both land and ocean data, with the hypothesis that the clustering algorithm will be able to identify different clusters both based on spatial location (i.e. land versus ocean) and temporal variability (i.e. stratus versus convective). To do this, we estimate that we will need 10-20 different clusters for all of North and South America in order to capture both the regional and temporal variability; however, to start we will attempt to find and interpret 5-8 clusters on the oceanic subset.

k-means + Autoencoder

An autoencoder, first introduced by Rumelhart *et al.* (1986) is a fully connected neural network with the ability to learn a compressed representation of data. Dimensionality reduction is performed by learning a latent vector with fewer units than the input and output. The model will be trained using an unsupervised approach with the raw GOES channels as the input and compared to itself as the target. To cope with the large quantity of data samples we will use the distributed PyTorch library for training.

Weights of the latent vector will parameterize the encoded input samples to produce new data points with fewer dimensions. The *k*-means clustering algorithm will be applied over these new points to identify the underlying characteristics. A centralized approach to clustering, using Python, will prove adequate with the decrease in data size. We hypothesize that different cloud regimes will fall into respective clusters.

Deep Embedded Clustering Algorithm

Finally, Deep Embedded Clustering (Xie *et al.,* 2015) consisting of a pre-trained autoencoder and a clustering layer will be implemented on the satellite data in order to simultaneously improve the feature representation and the clustering of different types of clouds. Depending on the complexity of the model and the time availability, convolutional autoencoder will be tried in case a fully connected deep neural network fails to exhibit satisfactory performance. The result will be compared with both the baseline *k*-means result and the combined *k*-means + Autoencoder result. PyTorch will be used for the implementation of Deep Embedded Clustering on distributed platform.

[2] Apache Spark Clustering Documentation https://spark.apache.org/docs/latest/ml-clustering.html

Evaluation

Performance of clustering and the supporting neural networks will be evaluated in our approach. We will use clustering accuracy such as silhouette coefficient when ground truth labels are not known. Depending on the class balance in the dataset, we might also use the area under the ROC curve. Root-mean-square error (RMSE) will provide insight to how the neural networks compare. Multiple architectures will be explored with varying layer sizes and latent vector dimensions.

The different clusters will be analyzed and compared to expected meteorological regimes using both the cluster centroid statistics and maps of the clusters predicted across the spatial domain at different times. Specifically, we will focus on if the clustering algorithms are able to detect different regional cloud regimes as well as common regimes, such as low clouds. If time permits, this will also be evaluated against low cloud presence from the CloudSat and CALIPSO satellites, for which we have co-located data for their overpass tracks.

We will also analyze the three approaches with respect to training and run time, and how these scale with the number of machines. Since the real-world data comes down from the satellite every 10 minutes, if our approach were to be put into production, it would need to be able to scale efficiently and process the data within that time. By extrapolating from runs on several different cluster sizes, we will estimate how many machines (and which algorithms) would be necessary to keep up with the full data stream.

Timeline

Mar 30	Setup GitHub repository. Provide an overview of the dataset and begin preprocessing. Review existing literature and software documentation for libraries that we plan to use.
Jason Stock	Create a GitHub repository with folders with default Spark projects and scripts, as well as basic distributed learning using PyTorch.
Katherine Haynes	Gather and preprocess data.
Nazia Farhat	Analysis of the satellite data, assist in data preprocessing, and work on algorithm design
Joe Strout	Begin work on basic *k*-means code (perhaps on smaller artificial dataset).
Apr 6	Start design of algorithms for each of the aforementioned approaches.
Jason Stock	Implementing an autoencoder with the dataset. Investigate different dimensions of latent vector for clustering. Capture and contrast RMSE of various neural network architectures to find the best models. Generate datasets from encoding samples to begin clustering.
Katherine Haynes	Assist with *k*-means clustering and Deep Embedded Clustering.
Nazia Farhat	Formulation and implementation of distributed Deep Embedded Clustering algorithm
Joe Strout	Finish *k*-means clustering code applied to real data.
Apr 13	Evaluation of results and generating tables/figures that quantify our analysis.
Jason Stock	Illustrate how the neural networks compare to each other. Evaluate how different architectures influence clustering performance.

Katherine Haynes	Assist with evaluation and analysis.
Nazia Farhat	Analyze the findings of the project and work on visual/quantifiable representation.
Joe Strout	Outline project report.
Apr 20	Report and presentation preparation
Jason Stock	Assist in preparing project report and presentation.
Katherine Haynes	Assist in preparing project report and presentation.
Nazia Farhat	Assist in preparing project report and presentation.
Joe Strout	Assist in preparing project report and presentation.

Bibliography

Andersen, H. & Cermak, J. (2018). First fully diurnal fog and low cloud satellite detection reveals life cycle in the Namib. *Atmos. Meas. Tech*., **11**, 5471-5470, https://doi.org/10.5194/amt-11-5461-2018.

Biookaghazadeh, S., Xu, Y., Zhou, S. & Zhao, M. (2015). Enabling scientific data storage and processing on big-data systems. *IEEE International Conference on Big Data*, 1978-1984.

Chen, C., Cooley, D., Runge, J., & Szekely, E. (Eds.). (2018). *Proceedings of the 8th International Workshop on Climate Informatics: CI 2018* (No. NCAR/TN-550+PROC). doi:10.5065/D6BZ64XQ

Qin, Y., Steven, A.D.L., Schroeder, T., McVicar, T.R., Huang, J., Cope, M. & Zhou, S. (2019). Cloud cover in the Australian region: development and validation of a cloud masking, classification and optical depth retrieval algorithm for the Advanced Himawari Imager. *Front. Environ. Sci.*, **7**:20, https://doi.org/10.3389/fenvs.2019.00020.

Rumelhart, D. E.; Hinton, G. E. & Williams, R. J. (1986), Learning Internal Representations by Error Propagation, *in David E. Rumelhart & James L. Mcclelland*, ed., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundation;* MIT Press, Cambridge, MA, **318**:362.

Shange, H., Letu, H., Nakajima, T.Y., Wang, Z., Ma, R., Wang, T., … & Shi, J. (2017). Diurnal cycle and seasonal variation of cloud cover over the Tibetan Plateau as determined from Himawari-8 new-generation geostationary satellite data. *Scientific Reports*, **8** (1105), https://doi.org/10.1038/s41598-018-19431-w.

Xie, J., Girshick, R. & Farhadi, I. (2016). Unsupervised deep embedding for clustering analysis. *Proceedings of Machine Learning Research*, **48**, 478–487, New York, New York, USA, 20–22.

Wind, G., Platnick, S., King, M.D., Hubanks, P.A., Pavolonis, M.J., Heidinger, A.K., Yang, P., & Baum, B.A. (2010). Multilayer cloud detection with the MODIS near-infrared water vapor absorption band. *J. Appl. Meteor. Climatol.*, **49**, 2315-2333. https://doi.org/10.1175/2010JAMC2364.1.