Lab 2 - Cloud Data
Stat 214, Spring 2025

These instructions are specific to **Lab 2** and cover the background and deliverables needed to complete the assignment. Please also see the general lab instructions in **discussion/week1/lab-instructions.pdf** for information on how to set up your environment, write the lab report, and submit the final product.

Contents

1 Submission

Push a folder called `lab2` to your `stat-214 GitHub` repository by **23:59 on Friday, March 21st**. Although this is a group project, each group member must submit the group's report and code in their own repo. We will run a script that will pull from each of your GitHub repositories promptly at midnight so take care not to be late as late labs will not be accepted.

The 12-page limit is not in place for this lab. The page limit for this lab is 20 pages. The bibliography and academic honesty sections don't count towards this limit.

Follow the general lab instructions in stat-214-gsi/discussion/week1/lab-instructions.pdf for more details. Please do not try to make your lab fit the requirements at the last minute!

We have provided a `.tex` template as a guideline for the writeup. Since the template is intended to make grading easier, please do not deviate from it significantly without good reason. (We have not provided a `.ipynb` template because collaboratively writing this report in a notebook is a bad idea). In your `lab2` folder, please provide everything (except the data) that is needed for someone else to be able to compile your report and receive the exact same pdf.

Special coding instructions

For any autoencoders you train, please save the trained model (as a model checkpoint) in your `results` directory. It should be a `.pt` file. If you use YAML files to configure training runs (hint: you should), please include them somewhere in your `code` directory.

2 Academic honesty and teamwork

Academic honesty statement

We ask you to draft a personal academic integrity pledge, addressed to Bin, that you will include with all of your assignments throughout the semester. This should be a short statement, in your own words, that the work in this report is your own and that all sources you used are properly cited, including your classmates. Please answer the following question: Why is academic research honesty necessary? If you feel it is not, make a clear argument why not.

Collaboration policy

Within-group: In a file named `collaboration.txt` in your `report` directory, you must include a statement detailing the contributions of each student, which indicates who worked on what. After the labs are submitted, we will also ask you to privately review your group members' contributions.

With other people: You are welcome to discuss **ideas** with the course staff or other students, but your report must be written up and completed by your group alone. Do not share code or copy/paste any part of the writeup. If you discuss with other students, you must acknowledge these students in your lab report.

LLM usage policy

You are allowed to use LLMs (ChatGPT, GitHub Copilot, etc.) to *assist* in this lab, but are not allowed to use it for more than creating visualizations or helping correct grammar in the report. If we have reason to believe an LLM wrote a significant portion of your code (more than 5%) without your editing or iteration, or any section of your report word-for-word, this will constitute an honor code violation.

3 What you need to do

The goal of this lab is to explore remote sensing images and develop cloud detection algorithms in the polar regions using radiance data collected by the Multi-angle Imaging SpectroRadiometer (MISR) sensor aboard NASA's Terra satellite. The cloud mask (signifying where the clouds are) is an important input to climate models since clouds can keep the thermal heat near the earth surface (warming effect) and at the same time reflect sun radiation back to space (cooling effect). Satellite images provide useful climate information based on two main observations: (i) different objects of interest have different colors and (ii) measurements closer to the earth surface tend to be warmer or have higher temperature. These two main observations do not work for cloud detection over the polar regions because both clouds and polar region surface are white and cold. Hence, the MISR sensor was developed to take advantage of the fact that clouds are at a different altitude than the polar region surface.

Your task is to build a prediction model to distinguish cloud from non-cloud for each pixel using the available MISR images. You will be provided with 164 images, only 3 of which will have "expert labels" that you can use to train classifiers. The unlabeled images will enable you to experiment with transfer learning and pre-training to facilitate the classification task.

The data can be found in bCourses under `Files > Labs > lab2 > image_data.zip`, which contains all 164 images in `.npz` format. Each of these files contains one "picture" from the satellite. Each of these files contains the 10 columns described below. Images `O013257.npz`, `O013490.npz`, `O012791.npz` have an additional column containing expert annotations. NDAI, SD and CORR are features developed by researchers based on subject matter knowledge. They are described in the article `yu2008.pdf` in the `lab2/documents` folder. The sensor data is multi-angle and recorded in the red-band. More information on MISR is available at `http://www-misr.jpl.nasa.gov/`

0	y coordinate
1	x coordinate
2	NDAI
3	SD
4	CORR
5	Radiance angle DF
6	Radiance angle CF
7	Radiance angle BF
8	Radiance angle AF
9	Radiance angle AN
10	expert label (+1: cloud, −1: not cloud, 0: unlabeled)

The following are the instructions for the three main parts of the lab. Please carefully document your analysis pipeline, justify the choices you make, and place your work within the domain context. You should have an introduction and conclusion section in your report.

Part 1: Exploratory Data Analysis (10% of Lab 2 Grade)

1. For the 3 labeled images, plot the expert labels for the presence or absence of clouds according to a map (i.e. use the X, Y coordinates).
2. Explore the relationships between the radiances of different angles, both visually and quantitatively. Do you notice differences between the two classes (cloud, no cloud) based on the radiances? Are there differences based on the features (CORR, NDAI, SD)?
3. Split your data into training, validation, and test sets (or training and test, with some cross-validation scheme over the training set). Justify your choice of split based on how it reflects the challenges with possible future applications of your cloud detection algorithm.
4. As is common when dealing with real world data, these images may have imperfections that you'll need to clean up. Recall the skills you developed during Lab 1.

Part 2: Feature Engineering (30% of Lab 2 Grade)

1. Some of the features might be better predictors of the presence of clouds than others. Assuming the expert labels are accurate, use data analysis to identify and justify three of the most informative features. Provide both quantitative metrics and visualizations to support your selection. Note that while this analysis helps understand feature importance, you are not restricted to using only these features in your classification models.
2. Based on this information, can you engineer any new features? For example, the three features NDAI, SD, and CORR were created using expert knowledge, but they may not take full advantage of information contained in surrounding pixels. Perhaps you could create and try some new features that use a patch of data around a point?
3. Transfer learning: see next section.

Transfer Learning

Transfer learning is a machine learning topic that studies how well a model pre-trained on one task can be adapted to a different but related task. In our case, we will pre-train our autoencoder on the large set of unlabeled images before fine-tuning it on our 3 labeled training images. Note that even though we are fine-tuning it on labeled images, we do not touch the expert labels in this step. This kind of pre-training is an unsupervised learning problem that doesn't require labels. The "transfer" comes from the differences in cloud patterns, landmarks, and brightness between the unlabeled and labeled images.

In the script `lab2/code/autoencoder.py`, we have provided an autoencoder which takes small patches of points (9x9 squares with 8 features, by default), flattens them, and encodes them into a low-dimensional latent space. The script also includes code to train and evaluate the model. This code is rudimentary and intended to serve as a starting point. There are many ways to improve upon it.

To improve performance, experiment with different hyperparameters. We suggest trying a learning rate in the range of 10^{-3} to 10^{-4}, using Adam as your optimization algorithm, and starting with the default PyTorch initialization. Additionally, modify the architecture slightly (consider adjusting the number of hidden layers, units per layer, or activation functions) to see if you can obtain a better embedding that more clearly separates cloudy and non-cloudy pixels. (Note: the autoencoder should have the same input/output dimensions as the one we have provided). Use some of the techniques described in the discussion section to pick an appropriate latent dimension.

Part 3: Predictive Modeling (60% of Lab 2 Grade)

1. Develop several (i.e., at least three) 0-1 classifiers for the presence of clouds, using your best features and autoencoder features, or others, as necessary. Provide a brief description of the classifier, state the assumptions of the classification models if any, and test if the model assumptions are reasonable.
2. Assess the fit for different classification models e.g. using cross-validation, AIC, and/or the ROC curve. Think carefully about how to choose folds in cross validation and/or your training/test splits.
3. Pick a good classifier. Show some diagnostic plots or information related to convergence, parameter estimation (depending on your choice of models), and feature importance. The last item will be especially interesting after all of our effort spent engineering good features. Some examples of useful techniques are MDI+, coefficient magnitudes, permutation importance, etc.
4. For your best classification model(s), perform some post-hoc EDA. Do you notice any patterns in the misclassification errors? Again, use quantitative and visual methods of analysis. Do you notice problems in particular regions, or in specific ranges of feature values?
5. How well do you think your model will work on future data without expert labels? Is there a stability analysis, or some test that might give you a good idea of this? Investigate how your results change when you conduct various perturbations throughout your prediction pipeline. As a sanity check, run your classifier on some of the unlabeled images to see if the predictions look reasonable.

4 Peer Grading

So that you each have the opportunity to see a smattering of alternative approaches to the labs, we will be doing peer-grading for this class.

You will each receive 2 reports from your peers to grade. A detailed rubric will be provided and you will be expected to provide both written feedback as well as a numeric grade on a variety of topics including communication, quality of data cleaning, relevance of visualizations, and reproducibility (can you easily re-compile their report).

After you have all submitted your own assignments (and shortly after the deadline), we will run a script that will automatically push two randomly selected reports into a folder called `lab2/peer_review/`. To retrieve your allocated reports, you will need to `git pull`. You will have one week to review these two reports and return your feedback in the form of a Google questionnaire that we will send by email to you.

We will use these two grades for your report as a guide for grading, rather than as a final decision on your grade.