Project Proposal

CSCI 470/575: Introduction to Machine Learning

Mushroom Research and Development

Taqi Alyousuf

Arthur Clement

Madeline McKune

Austin Oltmanns

Anna Titova

Applied Project

Preference: High

Topic Area: Salt identification in geolocial images.

Project Name: TGS Salt Identification Challenge

Problem Statement: Kaggle description: Several areas of Earth with large accumulations of oil and gas also have huge deposits of salt below the surface. Unfortunately, knowing where large salt deposits are precisely is very difficult. Professional seismic imaging still requires expert human interpretation of salt bodies. This leads to very subjective, highly variable renderings. More alarmingly, it leads to potentially dangerous situations for oil and gas company drillers. To create the most accurate seismic images and 3D renderings, TGS (the worlds leading geoscience data company) is hoping Kaggles machine learning community will be able to build an algorithm that automatically and accurately identifies if a subsurface target is a salt body or not.

Proposed Solution: To develop our solution, we will first survey existing solutions to this problem. After analyzing existing solutions, we plan on utilizing the information learned in order to develop our own solution. This will consist of finding the best way to transform the input data (possibly no transformation) before applying a machine learning process which has parameters that can be trained to the given data to produce an output that may also be transformed to produce a final output. Because this dataset is so large and varied, a traditional signal processing/programming approach may fall short due to variations in the data and the range of potential inputs. For this reason, machine learning techniques will provide an advantage as they are able to learn how to respond to such wide and varied inputs.

Based on initial reasearch, some items we may investigate include various ML algorithms using supervised and/or unsupervised learning. Supervised learning includes Convolutional Neural Networks (CNN) and Feedforward networks (FFN) with multiple hidden layers. The role of unsupervised learning such as K-mean clustering will be investigated. A combination of supervised and unsupervised could provide more value by means of deep belief networks. Our approach could extend to test different data attributes that could reveal extra features.

We will explore the use of CNN to identify salt from sediment before we design our model because CNNs are classically used for image classification problems. Identifying salt from sediment takes a trained eye and cannot be identified easily by the average human. By using Machine Learning, we will be able to compensate for these properties of the data which would not be as readily possible using traditional techniques.

Our project will take in a 101x101 pixel, geographic image and we will produce a mask that classifies each pixel as either salt (white) or sediment (black). The performance of the developed solutions will be evaluated by developing a confusion matrix for each solution. Accuracy metrics will then be used to ascertain the best solution.

Data: The data will be collected from the TGS challenge from Kaggle. The data includes a train and test folder, and a file with the depth each image was taken at. Inside the train folder, there are approximately 8,000 raw geographic images and their corresponding masks. A mask has sediment regions as black and salt as white. The test folder has approximately 18,000 raw geographic images and the model must learn the mask. Data provided by https://www.kaggle.com/c/tgs-salt-identification-challenge/data.

Applied Project

Preference: Low

Topic Area: Classification

Project Name: Mushroom Toxicity Estimator (MTE)

Problem Statement: Identifying mushrooms requires an understanding of fungi and their macroscopic structure. Given a dataset of 23 species of mushrooms, identify each as definitely edible (e), definitely poisonous (p), or unknown, from their attribute information.

Proposed Solution: We will first analyze two previously proven models for this dataset so that we will have a better understanding of the problem. We will either build upon our understanding of the model so that we can improve their solution or build our own model so that we can outperform their results. We plan to use Logical Regression because the algorithm best predicts the relationship between two variables from data. Logical regression will be applied to determine whether a mushroom type is poisonous or edible. The information for a single mushroom contains 23 columns of attributes and multiple classifications for each attribute. For example, the stalk-root for a mushroom can be bulbous=b, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r, missing=?. No human would be able to memorize every combination of attributes that corresponds to a poisonous or edible mushroom. Nor, predict which attributes are important for poisonous or edible groups. By using Machine Learning, this process is ameliorated because the model learns which attributes are important for each classification and can predict the toxicity for a given mushroom. We will take a table of 23 attributes (cap-shape, veil-type, etc.) and a classification for each attribute (broad, narrow, brown, etc.) and classify the mushroom as poisonous (p) or edible (e).

Data: The data will be collected from the Mushroom Classification challenge from Kaggle. The dataset includes a mushrooms.csv file which is a table of mushrooms by attributes. There are 23 columns of attributes that describe a feature of the mushroom. https://www.kaggle.com/uciml/mushroom-classification

Timeline

Work Breakdown Structure (WBS)

1. Project Proposal
   1.1. Task outline (TA)
   1.2. Project 1 solution description (CA)
   1.3. Project finding (MM)
   1.4. Project 2 solution description (AO)
   1.5. Final assembly of proposal (AT)
2. Analyzing existing algorithms
   2.1. Investigate existing algorithms (TA)
   2.2. Implement solution 1 (CA)
   2.3. Implement solution 1 (MM)
   2.4. Implement solution 2 (AO)
   2.5. Implement solution 2 (AT)
3. Outline our own method
   3.1. Define algorithm steps (TA)
   3.2. define step 1 (CA)
   3.3. define step 2 (MM)
   3.4. define steps 3, 4 (AO)
   3.5. Analyze proposed solution (AT)
4. Our own method generates results
   4.1. Implement step 1 (TA)
   4.2. Implement step 2 (CA)
   4.3. Implement step 3 (MM)
   4.4. Implement step 4 (AO)
   4.5. Test and validate (AT)
5. Our own method works acceptably
   5.1. Refine step 1 (TA)
   5.2. Refine step 2 (CA)
   5.3. Refine step 3 (MM)
   5.4. Refine step 4 (AO)
   5.5. Test and validate further (AT)
6. Outline presentation
   6.1. Introduction (TA)
   6.2. Method (CA)
   6.3. Method (MM)
   6.4. Conclusion (AO)
   6.5. Further work (AT)
7. Draft presentation
   7.1. Introduction (TA)
   7.2. Method (CA)
   7.3. Method (MM)
   7.4. Conclusion (AO)
   7.5. Further work (AT)
8. Presentation
   8.1. Introduction (TA)
   8.2. Method (CA)
   8.3. Method (MM)
   8.4. Conclusion (AO)
   8.5. Further work (AT)

Critical Path The critical path of the project with expected completion dates of each task is

- 1.0 Project Proposal (ALL) - (09/06/19)
- 2.0 Analyzing existing algorithms (ALL) (9/23/19)
- 3.0 Outline our own algorithm (ALL) (10/10/19)
- 4.0 Our own method generates results (ALL) (10/21/19)
- 5.0 Our own method works acceptably (ALL) (11/04/19)
- 6.0 Outline presentation (ALL) (11/11/19)
- 7.0 Draft of presentation (ALL) (11/18/19)
- 8.0 Presentation (ALL) (12/05/19)