Identifying Migration Arifacts and Reflectors by Machine Learning Methods

Yuqing Chen

May 3, 2018

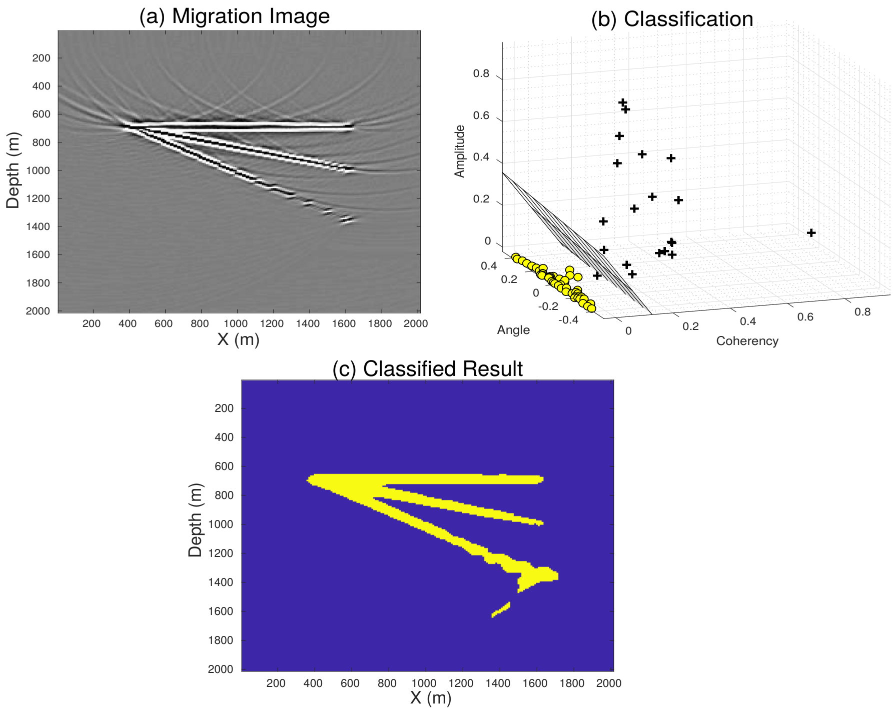

Figure 1: (a) Migration image of a three layer model with some artifacts which caused by sparse geometry, (b) classifiction results of training set and (c) classification results of all imaging points in the migration image.

1 Objective:

In this lab, we will go through three machine learning algorithms (logistic regression, support vector machine and nural network) to solve a same problem, which is trying to distinguish artifacts and reflectors in a migration image.

2 Required Software:

Matlab

3 Procedure of Machine Learning:

(1) Find features with dimension N. In our case, we choose the coherency, dipping angle and amplitude as our features, so N = 3.
(2) Build training sets. Manually pick 50 artifact points in the image and labled as 0, then manually pick 20 reflector points in the image and labeled as 1.
(3) Use machine learning algorithms to estimate a optimal N-1 dimension boundary which seperates the training set. In our case, the boundary is a 2D surface.
(4) Input the rest of points in the image into the trained model to do classification.

4 How to Use the Lab

(1) Download Comparision.tar and unzip it. Change your Matlab working diectory under this file so you may able to use all necessary sub-functions. The main function is **compare_main.m**, open it in the matlab script and run.
(2) The function **costFunction2.m** and **nnCostFunction.m** are used to compute the grdient and misfit of the logistic regression and neural network, respectively. For the SVM, the function **svmTrain.m** is used to train the model parameters. This function is in courtesy of Prof. Andrew Ng from his Machine Learning course from Coursera (https://www.coursera.org/learn/machine-learning/home/week/7).
(3) Load the migration image and corresponding 3 features: *cohe*, *angl* and *engy* represent the coherency, dipping angle and the amplitude, respectively.
(4) As different features has different unit, we rescale them into the range between -1 to 1 by:

$$\tilde{x} = \frac{x_i - \bar{x}}{\sigma}, \tag{1}$$

where, $\bar{x}$ is the mean of the feature and σ is the difference between the maximum value and minimum value in the feature.
(5) Manually pick 50 artifact points in the image and labled as 0, then manually pick 20 reflector points in the image and labeled as 1 to build the training set.
(6) Randomly initalize the initial model parameter and then compute the gradient and update the model parameter by Steepest descent method.

5 Logistic Regression

For a training sets with m examples, the misfit function of logistic regression can be written as:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}log(h_\theta(x^{(i)})) + (1 - y^{(i)})log(1 - h_\theta(x^{(i)}))], \tag{2}$$

and

$$h_\theta(x^{(i)}) = \frac{1}{1+e^{-\theta^T x}}, \quad (3)$$

where, x is the input features (3x1 vector) and y is the corresponding outputs (either 0 or 1). θ is the model parameters which are updated by least square iterations:

$$\theta_{k+1} = \theta_k - \alpha\frac{\partial J(\theta)}{\partial \theta_k} \quad (4)$$

.

6 Neural Network

For a training sets with m examples, the misfit function of the neural network can be written as:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}[y_k^{(i)} log(h_\theta(x^{(i)})_k) + (1-y_k^{(i)})log(1-h_\theta(x^{(i)})_k)], \quad (5)$$

where K is the number of layer in the neural network. We can see that if the architecture of neural network only have one layer (which has one input layer then followed with output layer), then the misfit function of the neural network is same as the logistic regression.

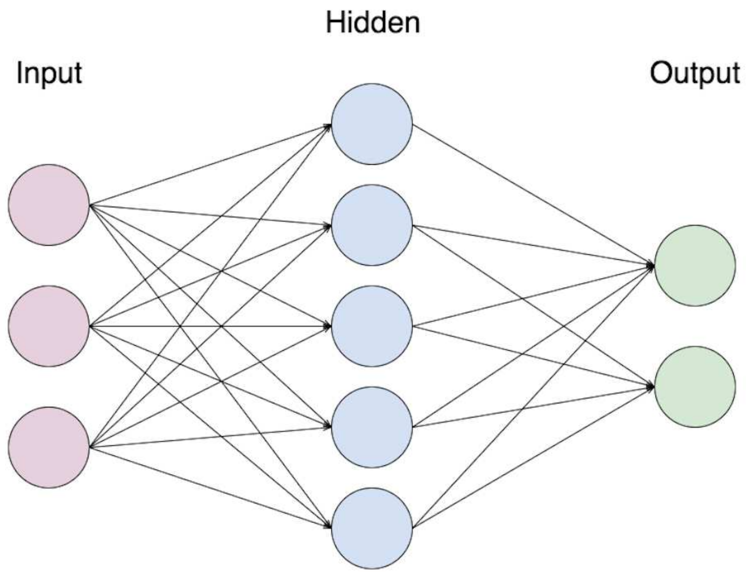

Figure 2: Neural network architecture with one input layer, one hidden layer and the output layer.

7 Support Vector Machine

The theory of support vector machine (SVM) is a bit of different compared with the logistic regression and neural network. SVM trying to separate different class of data with a clear gap that is as wide as possible.

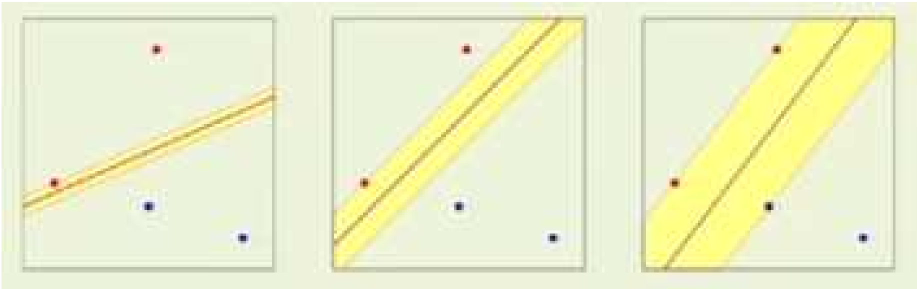

Figure 3: SVM trying to separate different class of data with a clear gap that is as wide as possible.

$$\begin{aligned} min_{\theta,b}\ d = \frac{1}{2}\theta^T\theta \\ sujet\ to\ y^{(i)}(\theta^T\mathbf{x}^{(i)} + b) \geq 1, \end{aligned} \quad (6)$$

where the parameter θ and the interception term b controls a bounday that seperates different class of data. This equation can be tranformed into a dual problem to solve for the optimal θ and b.

8 Questions

(1) Try to increase the freedom of the features from 1 to 3, or even higher orders for the logistic regression and see what the hyperplane will be looks like?
(2) Try to play with the number of layers and number of units in Neural network and see what the hyperplane will be looks like.
(3) Instead of using the skeletonzied information of the image, try to just use the intensity information of the migration image as the input, how will the ML algorithm perform? (Hint: for each image point in the migration image, open a local window with size of winx by winz around this image point, so in this case, the dimension of input features are winx by winz).
(4) If the number of Neural network only has one layer, what is the difference between one layer NN and logistic regression?

9 PS

Please let me know if there are any errors in this Lab, please contact: yuqing.chen@kaust.edu.sa

Regards,
Yuqing Chen