Rochester Institute of Technology, Computer Science Department

# Statistical Machine Learning

## Final Project

Vrishabh Ajay Lakhani

May 9, 2018

This problem is a very unique problem which combines two major problems which are hard to tackle using traditional approaches to machine learning, these are:

1. Working on unbalanced dataset
2. Working on very small dataset

The problem with such problem is that they tend to overfit the data or a part of data unless we have done sort of excellent feature engineering. Secondly, directly applying algorithms like a Neural Network might not converge at all or might overfit the data a bit too much. Almost as if it "memorizes" the each of the samples.

Thus, for the given project, I try a variety of machine learning problems which have proven to give really good results in such problems. The results are good and the model perfectly fit the data. I will also discuss all the possible approaches which I would have used just in case I hadn't got good enough results with just these many efforts. We will discuss each section of the data science pipeline individually for this write-up.

**Data pre-processing:**

Since all the images are of different sizes, we first normalize all the image to a particular size, that is, we resize all the images to one size before feeding them to the models.
Apart from this, we create a $5^{th}$ class which consists of random images downloaded from the internet which are used to identify images which aren't hand gestures.
We realize the size of the dataset is really small and thus we need to simulate the dataset and make it bigger. Thus we use Keras Image generator and apply transformations like width shift,

height shift, horizontal flip over the dataset. We don't use vertical flip because that might change the meaning of the gesture in American Sign Language.
Note: You can see the example transformed image in the "preview" folder.

***Possible improvements:***

1. We can add noise to the images while simulation to give better results on the models.
2. Secondly, we might get a better result if instead of using random images for the $5^{th}$ class we use random white noise, since random images might have some abstract similarity with the hand gestures and it might give us false positives especially since the dataset is small.

**Feature Engineering/ Representation learning:**

After pre-processing, we need to feature engineering the data before applying the classifiers onto the data. Here, we are doing it differently for different classes of the dataset.

For the neural network, since we are using convolutional neural network layer, it is important to input the entire image to the classifier since it needs to understand different patterns in the image.

While in case of non-neural network based classifiers, we need to feature engineer the data using various basis functions, apply feature selection algorithm and also apply feature decomposition algorithm to reduce the time complexity.

I am going to discuss only the algorithms which gave the best results here. First I flatten the image so that I get a 1-D vector of 30000 features (because the image size is 3x100x100). For the feature selection process, the best results I got were using SVM with a Linear Kernel. After feature selection, our input size reduces from 1x30000 to about 1x13500. This is still quite heavy for processing and converging thus we use feature decomposition in order to further reduce the size. According to the NIPS 2003 feature selection challenge, the best results were obtained using PCA, thus I apply PCA and finally get inputs of size 1x1000.

***Possible improvements:***

We can apply polynomial feature expansion over the input flattened image before feature selection. But it is a tradeoff between computation and accuracy.

**Classification:**

We use the K-Fold cross-validation to train the dataset.

***Possible improvements:***

We should use Stratified K-Fold cross validation whenever we want to deal with unbalanced dataset. I did not use it because I was already getting good results.

I try the following algorithms for classification:

1. Gradient Tree Boosting with regularization (XGboosting)
2. Adaboost classifier with SVM classifier (we notice that we get better results with a polynomial kernel as compared to RBF, just like the NIPS 2003 challenge results.
3. RandomForestClassifier with number estimators as 100:

We get the following results:

Accuracy:

1. Neural Networks: **100%**
2. Gradient Tree Boosting with regularization (XGboosting): **96%**
3. Adaboost classifier with SVM classifier: **65%**
4. Random Forest Classifier: **94%**

Time Complexity (Fastest first):

1. Random Forest Classifier
2. XGboosting
3. Adaboosting with SVM
4. Convolutional Neural Network (faster to train since I trained in on the GPU)

***Further Improvements:***

1. Pseudo Labelling: While training neural network, studies have shown to give better results if use pseudo-labelling the test set and training over it. That means, we predict the outcome of the test set, concatenate the output according the these target to the training set and retrain the network. It boosts the performance,
2. Using pre-trained Neural Network: We can use pre-trained Neural Network which are trained on the ImageNet and just retrain the last dense layer of the network. Thus it can take advantage of the bigger dataset to understand the patterns in object it learnt using a much larger dataset and we can use it to train on smaller datasets.
3. Density estimation using VAE: Instead of applying transformation to the training dataset. We can apply Variational Autoencoder to the dataset and generate new images which are based on the lower bound of the log-likelihood of each class. This can further improve the training.
4. RandomSearch: We should apply Random Search in order to hypertune all the parameters in our classifiers.
5. We can use Bayesian Neural Network along with this too.
6. We can use ensemble concepts like Stacking to further increase accuracy.

**Note:** Please keep the test image inside the “test” directory according to their classes.