Image Classification Project

Minji Pack

Andrew ID: mpack
Masters in Biotechnology
Innovation and Computation
2016-2018 Batch

1 Introduction

In traditional supervised machine learning, the more (labeled) data you can collect, the more accurate your prediction would be. However, in real life, it is often hard to collect as many labeled data as possible. This is mainly because the budget of work is limited, but querying label to Oracle(usually human oracle) is expensive or time-consuming tasks. This project also demonstrates the typical real-life problems by giving us the constraints of 2,500 calls as our total budget.

As a solution of such constraints, i.e., limited budget, active learning is a way for models to query labels they want and increase the accuracy of predictions. The key benefit of active learning is with fewer labels, it performs well or even better than the traditional way. I will describe my active learning approach with details below.

Constraints: We have 2,500 calls as our total budget.

2 Environment

I have used python as a programming language. For external libraries, **numpy** and [2]**scikit-learn** are used since [3]*numpy* is a powerful library to work with matrix, similar to *matlab*. *scikit-learn* has multiple machine learning models and useful utility methods you can benefit from. For example, I have utilized **predict_proba** from *scikit-learn*; this gives you the probability distribution indicating how likely each label can be assigned to the instance. This is used for uncertainty sampling.

3 Data Description

There are three data sets: Easy, Moderate, and Difficult data set. Each data set is also divided into three parts: train set, test set, and blinded set; the shape of each from Easy and Moderate data set is 4120 by 27 matrix. 1000 by 27 matrix, 250 by 27 matrix, whereas that of each from a Difficult data set is 4120 by 53. The row of the matrix represents the number of images and the column of the matrix represents the features taken into considerations. The number of labels, representing subcellular localization patterns, in the whole data set is 8:

1 Endosomes
2 Lysosomes
3 Mitochondria
4 Peroxisomes
5 Actin
6 Plasma Membrane
7 Microtubules
8 Endoplasmic Reticulum

4 Approach Overview

My active learning strategy can be divided into three parts: **Data Access Strategy**, **Base Learner**, **Query Strategy**.
I also tried **Recursive Feature Elimination**.

I have implemented my logic in python with *numpy* and *scikit-learn*.

1 Read the *csv* files and convert them into *numpy* array format, so that I can work with matrix calculation easily.
2 **Data Access**: Select initial batch from the training dataset and query their labels to Oracle. [**Pool-based Sampling**]
3 **Base Leaner**: Used two models as my base learners and compare each of them with a random learner to find out which would be outperformed model and select it as the model I would use.
4 **Query Strategy**
   (a) For **Active Learning** way, get the probability distribution of how likely each label could be assigned to. Calculate entropy of each distribution(per each instance) and select the points having the highest entropies; its size is the number of batches. [**Pool-based Sampling, Uncertainty Sampling**]
   (b) For **Random Learning** way, select random points; its size is the number of batches, and query labels to Oracle. [**Pool-based Sampling**]
5 Measure the errors(1.0-accuracy)
6 Plot both errors of a random learner and an active learner per the number of queries.

4.1 Data Access Strategy

I use poolbased sampling[1] as a way to access the dataset. The reason why I choose the pool-based sampling over Stream-based Selective Sampling is that pool-based sampling is getting a certain large amount of data at once, but Stream-based Selective Sampling takes one data at a time. We already have a large pool of data points given and we can query their labels to Oracle. So, it would be more appropriate and efficient to use pool-base sampling.

There are some parameters we should take care of when working with pool-based sampling.

1 *Initial batch size*: The number of queries we request to Oracle for the first time. This is fixed in the initial step.
2 *uncertain sample/random sample size*: The number of queries we request for each iteration. This is fixed in the initial step, which means it won't change during the iterations.
3 *budget*: our constraints, the total numbers of queries we can request to our oracle.

I have tried several parameter settings to find out which setting is suitable for each data set.

4.2 Base Learners

Since we have 8 labels on our overall data pool, it is reasonable to select classifier models.

4.3 Query Strategy

This is the most important stage of active learning stages because this query strategy part actually makes active learner outperform other learners. When we draw some data from batch(I am using pool-based sampling[1]), we need to decide on which data points to query their labels to Oracle. The data points should be selected are those, which make our base learners confused to make predictions. I proceed using [1] Uncertainty Sampling. I choose entropy as a means of measuring how uncertain the data is. For more details, from the beginning, I mentioned that *preditct_proba* gives us the probability distribution of each instance, the probabilities each instance matches to each label(in our case, 8 labels). If the distribution follows a uniform distribution, we are uncertain which label should be assigned to such instance. Entropy evaluates this uncertainty per an instance; therefore, the higher value of entropy is, the more uncertain we are about the prediction.

4.4 Feature Selection

I used **Recursive Feature Elimination** to help to remove features which have noises. This is done by using the Recursive feature elimination technique(RFE) under scikit-learn. A Logistic Regression model trained is on the data, and the weights of the model are used to find features with noises. This is found to be useful for the 'difficult' dataset especially when using the Support Vector Classifier with the RBF kernel. The method followed is to perform RFECV during Uncertainty sampling iterations, and finally when doing predictions.

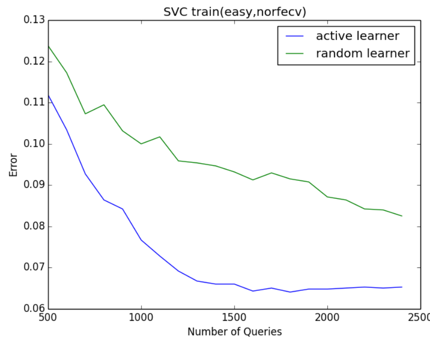
5 Experiments

5.1 Support Vector Machine

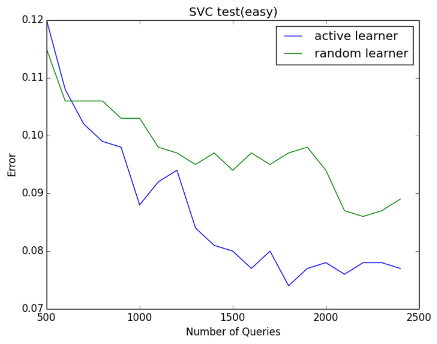
1 *initial batch size*: 500

2 *uncertain sample size/random sample size*: 100

5.1.1 Easy Dataset

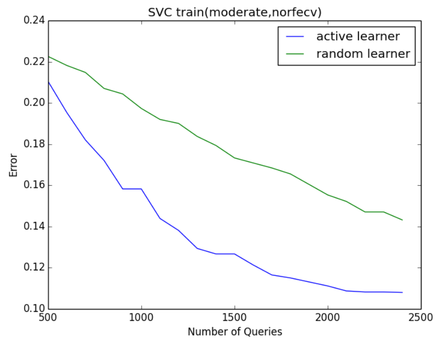


Train Error vs. Budget Size

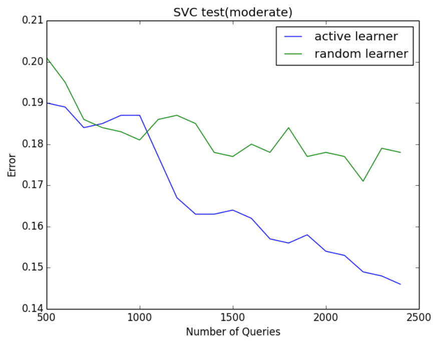


Test Error vs. Budget Size

5.1.2 Moderate Dataset



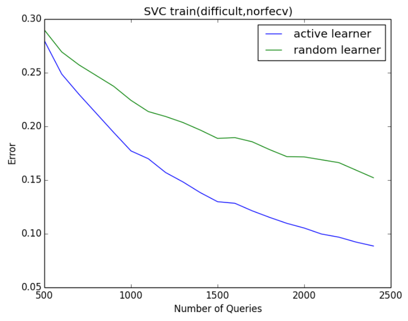
Train Error vs. Budget Size



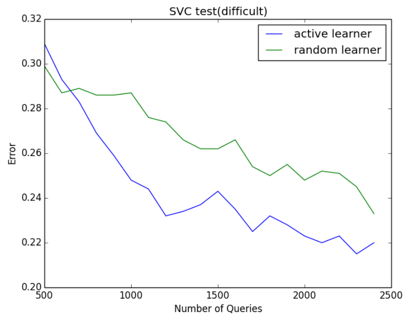
Test Error vs. Budget Size

5.1.3 Difficult Dataset

1 **No Feature Selection**:

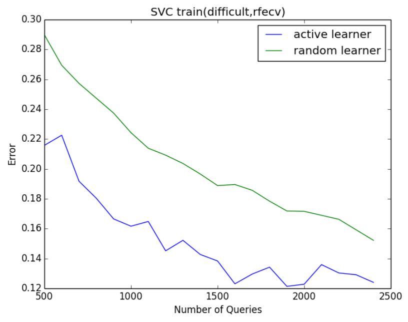


Train Error vs. Budget Size

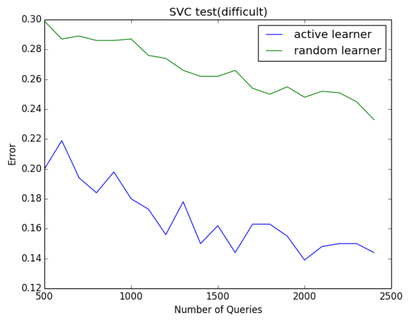


Test Error vs. Budget Size

2 **With Feature Selection**:



Train Error vs. Budget Size



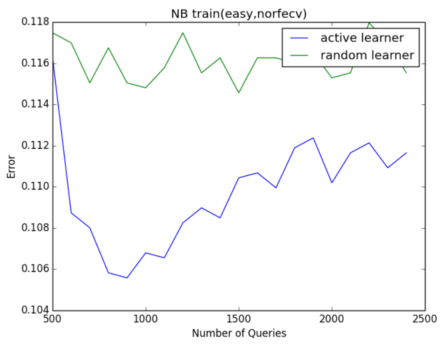
Test Error vs. Budget Size

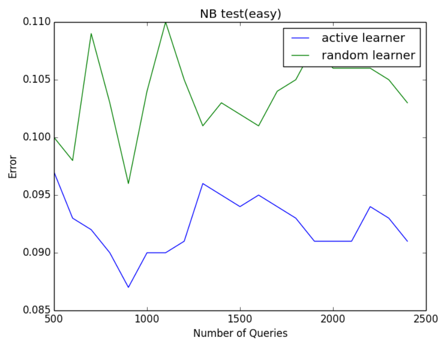
5.2 Gaussian Naive Bayes

1 *initial batch size*: 500

2 *uncertain sample size/random sample size*: 100

5.2.1 Easy Dataset

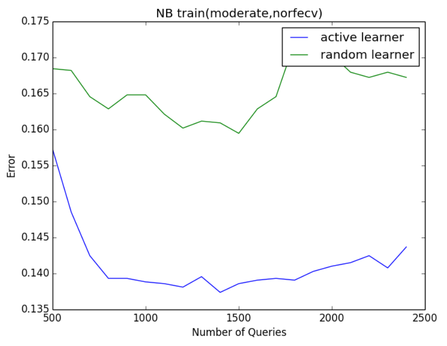


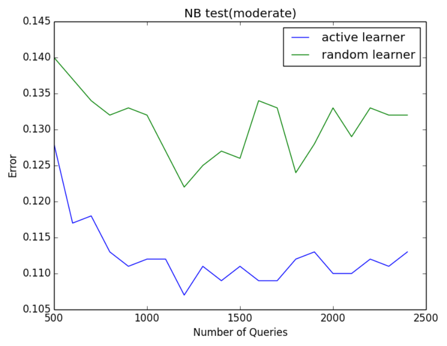


Train Error vs. Budget Size

Test Error vs. Budget Size

5.2.2 Moderate Dataset



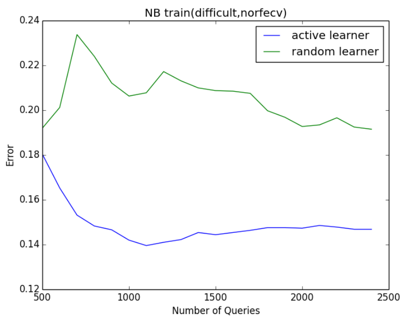


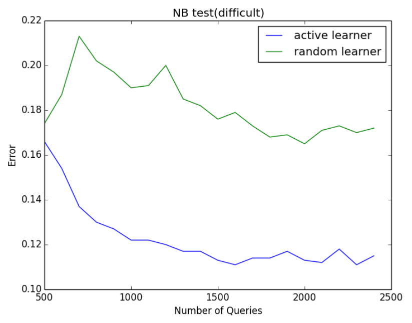
Train Error vs. Budget Size

Test Error vs. Budget Size

5.2.3 Difficult Dataset

1 **No Feature Selection**:

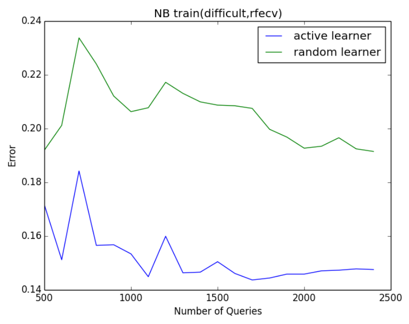




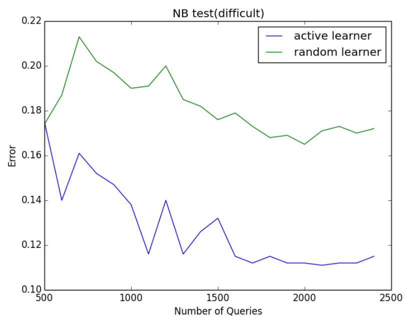
Train Error vs. Budget Size

Test Error vs. Budget Size

2 With Feature Selection:



Train Error vs. Budget Size



Test Error vs. Budget Size

References

1. *Burr Settles "Active Learning Literature Survey"*, (2009)
2. http://scikit-learn.org/stable/, Accessed 10th December, 2017
3. http://www.numpy.org/, Accessed 10th December, 2017