Analysis of Active Learning on Classification Models

Ishita Dasgupta
CICS, UMass Amherst
ishitadg@cs.umass.edu

Lopamudra Pal
CICS, UMass Amherst
lpal@cs.umass.edu

Sruthi Chilamakuri*
CICS, UMass Amherst
schilamakuri@cs.umass.edu

Abstract

Missing information can adversely affect prediction in machine learning models. Active Learning is a subfield of machine learning that solves this problem of learning missing data effectively and with lesser training[1]. It is the procedure of choosing the most informative data from the set of unlabeled instances that helps the machine learning model predict better. We study the effect of active learning on supervised learning models for classification problems. Our approach in this project was to exhaustively compare different aspects of Logistic Regression model, SVC model and Bayesian Inference model's classification performance. We implemented uncertainty sampling for each classifier under varying train and test conditions and found that Active Learning on Support Vector model performs better than that on Logistic Regression with increasing data. Although Active Sampling on Bayesian is better than Random sampling, yet the performance of Bayesian Active Learning is computationally intensive and its comparison with other classifiers can be affected by the optimum prior and sample posterior size selection. Our project approach is novel in a sense that it compares these very different classifiers on the top of multi-dimensional data complexity and performs a detailed analysis whilst stressing on the performance evaluation of rejection sampling on Bayesian Inference Active Learning model.

1 Introduction

Any supervised learning model requires hundreds or thousands of labeled data to be trained for efficient prediction but more often than not, these labeling can be expensive in terms of time, resources and effort required such as in speech recognition[2], information extraction[3] as well as in some classification problems. For better training our model when we do not have enough trained data, we can request for an oracle to provide us with labeled data from a pool of unlabeled instances. Active learning is an approach that does this efficiently by asking for most informative labeled data as compared to randomly asking for any labeled data. Thus, we target such a real-world classification problem that can be self-explanatory and relevant in demonstrating how Active Learning solves the problem of bottleneck due to missing unlabeled data.

1.1 Motivation

Active Learning is a relatively new approach whose effect on Classifying models is being exhaustively studied [3][7][13]. Our approach for the project is novel in the idea that we study a Bayesian model with Rejection Sampling on Active Learning as well as compare it to a Probabilistic and a

*Authors are in alphabetical order of their first names

Non-Probabilistic classifiers. There are different types of Active Learning approaches explained in [1]. We do a comparative analysis of time and prediction efficiency of pool-based active learning and batch-based active learning under Uncertainty Sampling approach since its the most commonly used framework. The real-world problem we imply this project on, is an assessment of health-risk for women belonging to developing countries. Over 60% of women in these countries face serious reproductive health problems such as sexually transmitted infections, unintended pregnancies, and complications from childbirth[5]. A survey was done to collect their personal and behavioral data in regards to their sexual and reproductive health. Data was collected across nine geographies to study the likelihood of getting such a disease or health hazard, which brings us to our classification problem. Active Learning is intuitive in such a problem due to difficulty in obtaining the data due to its sensitivity as well as probability of data mishandling.

1.2 Project Description

Thus, in our project we study Uncertainty Sampling[1] on our binary labeled dataset and study its effects on a MLE and MAP based Logistic Regression Model, SVM model, pool and batch-based Bayesian Inference on Logistic Regression model. Our choice of Active Learning and types of Classifiers explored is discussed in the Methodology adopted in Section 3. We perform hyper-parameter tuning on Datasets as described in section 4 within all model types via 1-fold cross validation and use the best results to compare across the classifiers whilst showcasing the benefits of active learning over passive learning. These experiments and results are detailed in section 5 and 6 respectively. Finally, section 7 concludes the project along with possible future work prospects.

2 Related Works

One of our main motivation behind doing this project was to analyze this comparatively new area of Active Learning that's been around not more than 10 to 15 years. [1] discusses different modes of Active Learning that can be applied on Supervised Learning Models, out of which we decided to work on Pool-based Uncertainty Sampling due to its ease of implementation and flexibility of application across models.
Our choice behind choosing the Classifiers to do a comparative evaluation was backed up by few of the recent works. [7] discusses the robust effectiveness of active sampling over random sampling on a logistic regression model. They run query-by-committee as well as uncertainty sampling techniques and amongst the various strategies tested, they find that active learning with widely used heuristic schemes almost always outperforms Passive learning cases. They also point out that Active Learning can be computationally as well as memory-intensive when performing basic binary classifications, although this varies with the dataset in question. They mention Computational complications being a bottleneck during evaluations which was true in our case as well. [8] was one of the first few papers to have demonstrated the use of Support Vector Classifiers with Active Learning by introducing the concept of version space for linearly separable data. Uncertainty Sampling uses maximum entropy method to find the least certain labeled data[1]. Since SVM is not a probabilistic classifier, it uses alternative methods that maximizes the margin considering both labeled and unlabeled data[8]. They discuss Simple Margin algorithm where the unlabeled data closest to the separating hyperplane is considered the least certain of being classified, but this heavily relies on the assumption that the version space is fairly symmetric. They alternatively come up with a MaxMin margin that tries to overcome these problems. [13] also discusses different query strategies for selecting informative data points with SVMs by comparing information-weighted Active Learning, Uncertainty Sampling, Online Learning, Batch-mode Active Learning, etc.
Extending the model to a Bayesian, [14] has provided the theoretical framework for parameter estimation whereas [10] implements a greedy active learning algorithm to choose from noisy observations. All these papers discuss our choice of classifiers individually but there is not enough work out there that compare these models. The comparison may not seem intuitive as they were designed for different purposes but it would be interesting to see how they perform under similar conditions giving us more insight into the working of the classifier models as well as the implications Active Learning has on these models.[15] implements Monte Carlo approach to reduce reduction in labeling error in future. They use entire pool of unlabeled data to estimate the expected error using Query-by-Committee technique. Their work is comparable to what we have tried to implement with Uncertainty Sampling but they use Naive Bayes as their model. [11] closely describes the approach

of selecting a batch of unlabeled data for manual labeling as opposed to a single unlabeled example to reduce the number of times the model must be retrained. This is an alternate approach to pool-based active sampling that we have implemented for some of our experiments.
We were also interested in testing classifiers under conditions they have not been tested under. For an instance, how an Active Learning model chooses labeled data from a pool of unlabeled dataset varies across classifiers as per their label-decision criteria. For example:- the probability of being on either side of the decision margin in case of Logistic Regression(LR) models compared to in [8], where idea of a version space in Support Vector Classifiers(SVC) is suggested for separating data that are linearly separable. We further want to extend this to test how LR and SVC performs on a non-linearly separable data and compared to each other. Another such scenario that we are interested is that in [9], the paper talks about a Gaussian Bayesian model that adaptively selects tests to perform when distribution's prior is known. In our case, we test on an unknown prior. We assume a Gaussian prior and test accuracies on varying prior to choose the optimal mean and covariance value.
Finally, we can sum up from our literature review that there isn't enough exhaustive work done on comparing various classifiers especially Bayesian to a Non-probabilistic and a probabilistic classifier. Novelty in our project also extends to the fact that our data is multi-dimensional and we test computational time and accuracies to study its effect on the classifiers. For our Bayesian Inference model, we have implemented Rejection Sampling for parameter estimation to study how all the models converge when the data amount is low.

3 Methodology

Active Learning is a sub-field of artificial intelligence and machine learning in which the active learning system develops and tests new hypotheses as part of the learning process. In contrast, passive learning systems induce hypotheses to explain whatever data is available.

Active Learning is useful when obtaining labeled data is difficult or expensive but unlabeled data is easily available and we would require additional labels to train the model accurately. Given that obtaining labels is expensive, the learner must choose intelligently which label to query next. A utility measure can be employed to measure the extent of uncertainty in the model's labelings and the most uncertain instance can be queried. The learner chooses to query the most uncertain instance because the label of that instance is most informative and more likely to affect classification. As we are measuring uncertainty and using it to influence our decision of selecting which label to query next, this method is termed uncertainty sampling. We present below the basic active learning algorithm with uncertainty sampling algorithm:

Algorithm 1 Active Learning with Uncertainty Sampling

procedure ACTIVELEARNING
    $U \leftarrow$ Pool of unlabeled instances
    $L \leftarrow$ Set of initial labeled instances
loop: t = 1,2,.....T do
    $\theta \leftarrow train(L)$.
    Select $x^* \in U$, the most uncertain instance
    Query the oracle to obtain label y^*
    Add x^*, y^* to L
    Remove x^* from U
    goto *loop*

In Algorithm 1, T represents the maximum number of queries the learner can ask. This algorithm is also called pool based active learning as in every iteration we are selecting the most uncertain instance from a pool of unlabeled data. One of the drawbacks to this algorithm is that the model must be retrained after every query. Depending on the type of model this can take very long.

A quick fix to minimize the number of times the model must be retrained is to use batching. Essentially, we select a batch of uncertain instances as opposed to a single instance for querying. Batch based active learning algorithm is presented in Algorithm2. Here, the total number of queries=T*batchsize; but the total number of times the model is trained is T.

Algorithm 2 Batch Active Learning with Uncertainty Sampling

procedure BATCHACTIVELEARNING
    $U \leftarrow$ Pool of unlabeled instances
    $L \leftarrow$ Set of initial labeled instances
loop: t = 1,2,.....T do
    $\theta \leftarrow train(L)$.

    loop: b = 1,2,.....batchsize do
    Select $x^* \in U$, the most uncertain instance
    Query the oracle to obtain label y^*
    Add x^*, y^* to L
    Remove x^* from U
    goto *loop*
    goto *loop*

We have used the entropy utility measure to quantify uncertainty in our active learning algorithm. Entropy is a measure of a variable's average information content. It is often thought of as an uncertainty or impurity measure in machine learning. The entropy utility measure is defined as:

$$X^* = \arg\max_x H_\theta(y|x)$$
$$= \arg\max_x \sum_y P_\theta(y|x) log P_\theta(y|x)$$

where, y ranges over all possible labelings of x.
More importantly, as long as the underlying model can provide a confidence or probability score, uncertainty sampling or the active learning algorithm can be implemented with the underlying model as a black box.
In our project, we aim to compare and contrast the performance of active learning across the classification models outlined in the following sections.

3.1 Logistic Regression (MLE)

We maximize the objective function

$$\arg\max_{w,b} \sum_{n=1}^{N} log \frac{1}{(1+exp(-(wx_n^T+b)))}^{[y=1]} \frac{exp(-(wx_n^T+b))}{(1+exp(-(wx_n^T+b)))}^{[y=0]}$$

3.2 Logistic Regression with Zero Mean Spherical Gaussian Prior (MAP estimate)

We maximize the objective function

$$\arg\max_{w,b} (\frac{w^T w}{2} + C \sum_{n=1}^{N} log \frac{1}{(1+exp(-(wx_n^T+b)))}^{[y=1]} \frac{exp(-(wx_n^T+b))}{(1+exp(-(wx_n^T+b)))}^{[y=0]})$$

where $C =$ Inverse of regularization strength

3.3 Logistic Regression with Bayesian Inference

$$Data\ D = (x_n, y_n)_{1:N}\ Prior P(\theta|\mu_0, \Sigma_0)$$

where,

Model Parameters $\theta = [w, b]$, $\mu_0 = Zero\ mean\ vector$, $\Sigma_0 = k*I$(k is a constant and I is the Identity matrix)

$$P(Y = y|X = x) = \frac{1}{(1+exp(-(wx_n^T+b)))}^{[y=1]} \frac{exp(-(wx_n^T+b))}{(1+exp(-(wx_n^T+b)))}^{[y=0]}$$

Parameter Posterior

$$P(\theta|D,\mu_0,\Sigma_0) = \frac{P(D|\theta)P(\theta|\mu_0,\Sigma_0)}{P(D)}$$
$$= \frac{P(D|\theta)P(\theta|\mu_0,\Sigma_0)}{\int P(D|\theta)P(\theta|\mu_0,\Sigma_0)d\theta}$$

$$P(D|\theta) = \prod_{n=1}^{N} P(Y=y|X=x,\theta)$$

As the evidence $\int P(D|\theta)P(\theta|\mu_0,\Sigma_0)$ is intractable, we use the Rejection Sampler[9] to draw samples from the parameter posterior.
Acceptance ratio for the rejection sampler is given by $\frac{P(x)}{MQ(x)} \leq u$, where u is drawn from Unif(0, 1)
For the rejection sampler $Q(x) = P(\theta)$, (samples are drawn from the prior) and $M = P(D|\theta^{MLE})$
Thus, acceptance ratio is $\frac{P(D|\theta^S)}{P(D|\theta^{MLE})} \leq u$

The Posterior Predictive distribution is given by

$$P(Y=y|X=x,D,\mu_0,\Sigma_0) = \int P(Y=y|X=x,\theta)P(\theta|D,\mu_0,\Sigma_0)$$

which can be approximated using Monte Carlo approximation as follows:

$$\frac{1}{S}\sum_{\theta\in\theta^{RS}} P(Y=y|X=x,\theta)$$

where, θ^{RS} are samples drawn from the parameter posterior using Rejection Sampling.
The predicted label probabilities for a new data point in Bayesian Inference would thus be

$$[\frac{1}{S}\sum_{\theta\in\theta^{RS}} P(Y=0|X=x,\theta), \frac{1}{S}\sum_{\theta\in\theta^{RS}} P(Y=1|X=x,\theta)]$$

We have used this probabilistic outputs to compute entropy in our uncertainity sampling utility measure. We have also contrast the behaviour of active learning for the following plug in estimate probability outputs. $\theta_{Mean} = [P(Y=0|X=x,\theta_{Mean}), P(Y=1|X=x,\theta_{Mean})]$ where $\theta_{Mean} = \frac{1}{S}\sum\theta_n^{RS}$

$\theta_{MAP} = [P(Y=0|X=x,\theta_{MAP}), P(Y=1|X=x,\theta_{MAP})]$
$\theta_{MLE} = [P(Y=0|X=x,\theta_{MLE}), P(Y=1|X=x,\theta_{MLE})]$

3.4 SVC

We use Simple Margin method to choose our most-informative unlabeled data or the point with maximum Entropy. We can approximate the version space efficiently in SVM in case of high-dimensional feature space. In the version space, the SVM solution w is the center of the hypersphere touching the hyperplanes induced by the support vectors. We assume the hypersphere is close to the center of the version space. A hyperplane now close to the center bisects the hyperspace. Therefore, we want to query the simple math formula that induces a hyperplane as close to w as possible.

In other words, this strategy queries the sample closest to the current decision boundary and is called Simple Margin. A detailed description of the tests we have conducted is outlined in the Experiments section.

4 Datasets Used

4.1 Data Description

We have used a real-world problem of Women's Health Risk Analysis to apply benefits of Active Learning in case of missing sensitive data labels. This dataset has been collected from a competition

that was organized to create a machine learning model which will automatically classify women into different health risk segments based on the information collected from the participants by Cortana Intelligence Gallery[4,5]. The data was provided in accordance with Bill & Melinda Gates Foundation open data access policy. Original data had 5000 training sample data of women aged 15 to 30 years old. There were 50 features describing personal belongings like cars, tv, etc that is indicative of their connectivity with the outside world. There are features indicating their literacy,education,geological information as well as personal marital and sexual health informational features. More information about individual features can be found in [5]

4.2 Data Preprocessing

Looking into the data, we found that only 1417 of them had complete data without any missing attribute values. We tried replacing the missing attribute values with 0 and using scikit-learn Imputer[6] over mean,median and most-frequent values, but the accuracy did not change much when tested with an SVM radial basis function kernel model. At the risk of biasing the data whatever approach we use to fill the missing NaN values, we went ahead with 1417 data samples, out of which for most experiments(unless mentioned otherwise) Train data was set to 1000 and test to 417 data points. We converted our dataset to a problem for binary class as opposed to a multiclass problem in [12] to avoid computational and time complexity. 3 of the 50 features were identifiers('religion', 'patientID', 'INTNR') and hence we got rid of them, resulting in 46-dimensional Data samples and 1-D labels. We also tried getting rid of some of the repetitive features such as 'urban' and 'urbanicity' and so on, to reduce the effect of multidimensionality but without much success. More on this is discussed in the Experimentation section 5.

5 Experiments

Experimental setup was as follows:

- Software or Programming language used: Python 2.7.6
- Existing Libraries used: Scikit-learn, numpy, pandas

5.1 Effect of Active Learning on Logistic Regression(LR) and Support Vector Machines(SVMs)

We first study how Active Learning works on Non-Bayesian Models such as LR and SVC. We apply pool-based Active and Random Sampling on the classifiers. We fix the SVC regularization parameter to 1, LR regularization parameter to 5 and a rbf kernel is used for the SVM to classify our multidimensional data. For this, we use the train-test data as described in the Dataset section and study the following parameters:

- Accuracy: The queries to the unlabeled dataset are in increments of 100 and the accuracy of the models after addition of a set of randomly chosen and actively chosen queries are reported.
- Runtime: Time taken by both the classifier models to apply active learning over an increment of 100 queries is reported.

5.2 Active Vs. Passive Learning on Bayesian Inference Logistic Model

In this section, we compare how Random Sampling on a Bayesian Logistic Regression Model works on Active and Randomly selected data labels from unlabeled data samples. Due to computational complexity of pool-based active learning, we perform these experiments on Batch-Based Uncertainty Sampling. The prior is assumed to be Gaussian and is set to 5. Regularization for the Logistic model is set to 1e33. Since Bayesian is very time consuming, the train and test data size are each set to 100. The sample drawing size for Random sampling is set to 100.

5.3 Comparison of All Classifiers on small amount of data

This is a detailed comparison of Active learning with increasing queries on the following models:

- Logistic Regression MLE: We set C to 1e12
- Logistic Regression MAP: We set C to 5
- SVC: C is set to 1, kernel = radial basis function
- Pool-based Bayesian: Over a space of 100 posterior parameters
- Pool-based Posterior: Over mean of the posterior

5.4 Batch vs. Pool-based Uncertainty Sampling

We compare these two types of Active Learning to study their computational(runtime) complexity and prediction performance over a set of train-test data of size 100 each as Bayesian takes a long time to converge for more data.

- Accuracy: The queries to the unlabeled dataset are in increments of 100 and the accuracy of the models after addition of a set of randomly chosen and actively chosen queries are reported.
- Runtime: Time taken by both the classifier models to apply active learning over an increment of 100 queries is reported.

5.5 Hyper-parameter Tuning

All experiments for hyper-parameter tuning on Bayesian Active Learning was done on Batch-based model because pool-based Active learning Bayesian model was a resource bottleneck. We perform the following hyper-parameter tuning on all our models via 1-fold cross validation:

- Change the regularization parameter on increasing queries for Active Learning on Logistic Regression model.
- Change the regularization parameter on increasing queries for Active Learning on SVC.
- Find the effective prior for random sampling in Bayesian Inference model with increasing queries for Active Learning
- Observe the effect of increasing posterior sample draw size in random sampling on Bayesian Inference model with increasing queries for Active Learning
- Observe the effect of increasing Unlabeled sample set size increasing queries for Active Learning on Bayesian model. Number of queries was fixed to a pool size of 50.

5.6 Objective Summary

In other words, the above experiments were performed to study the following analyses that can be summarized categorically as follows:

- Comparison of Logistic and SVM on large data: How non-bayesian classifers perform since with any amount of increasing data, their accuracy or performance is comparable as opposed to Bayesian frameworks for which it becomes difficult to compare performance for larger dataset.
- Comparison of all models on small data: Performance Comparison of all the models over a small dataset as well as performance of all posterior samples compared to posterior mean with increasing queries
- Study of Bayesian Model: Exhaustive understanding of when batch-bayesian should be performed and when pooling should be applied along with their overheads. We also study the effect of prior, posterior sample size and effect of choosing from an increaing set of unlabelled data and finally the performance of active and random sampling.

5.7 Hypothesis to justify

These experiments were performed to justify these hypothesis of our probable outcomes:

- Comparison of Logistic and SVM on large data: We expected the SVM to perform better because it uses an rbf kernel that can more accurately classify a multidimensional data.
- Comparison of all models on small data: Bayesian is expected perform the best with low data information when learned over all posterior sample spaces compared to all other, unless the sample size or prior assumption is not efficient. It is supposed to perform better than the posterior mean since it holds more information.
- Study of Bayesian Model: We were expecting Active Learning to perform better than Random Sampling and accuracy to increase with posterior sample size. Batch-bayesian was supposed to run faster as compared to pool-based but pool-based was supposed to return better accuracy.
- Other: Logistic Regression was supposed to perform better with low C for small data and SVM would slowly start to learn better with increasing data points due to its margin property.

6 Results

6.1 Effect of Active Learning on Logistic Regression and Support Vector Machines

6.1.1 Accuracy

Logistic Regression MAP Active Learning performs better on less data as compared to Support Vector Machines because of higher regularization strength. With more data-points for multidimensional feature Support Vector Machines with RBF kernel showed better results.

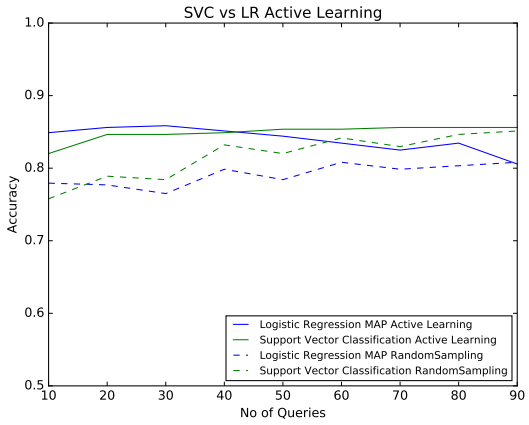

Figure 1: Effect of increasing queries on Accuracy for Active Learning LR and SVM

6.1.2 Runtime

Since the data is multidimensional and we are using RBF kernel, Support Vector Machines take more time as compared to Logistic Regression as be viewed in Figure2.

6.2 Active Vs. Passive Learning on Bayesian Inference Logistic Model

Figure3 shows that as we increase the number of queries the accuracy for Bayesian prediction increases. More importantly, the graph for active learning consistently shows improved performance as compared to passive learning. This result is as expected due to the nature of intelligent labeling of unlabeled data during active learning.

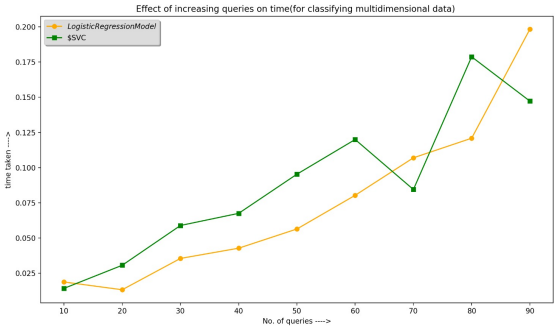

Figure 2: Effect of increasing queries on Runtime for Active Learning LR and SVM

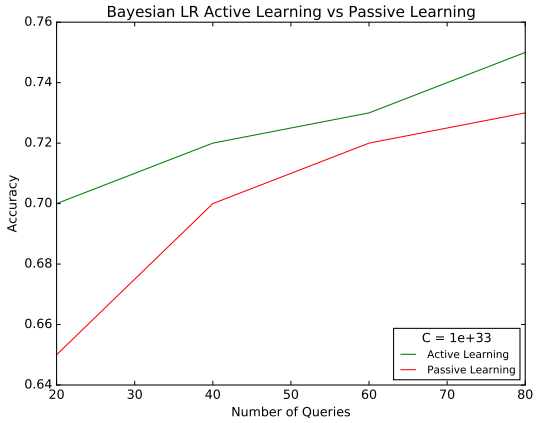

Figure 3: Comparing Active vs Passive Learning on Bayesian Logistic Regression

6.3 Comparison of All Classifiers on small amount of data

Figure 4 shows the comparison of Active Learning on all classifiers with increasing number of queries. The expected result was that Bayesian Active Learning will perform better than SVC Active Learning and that in turn will perform better than Logistic Regression since Bayesian is mostly applied on problems which have less data cases. However, this did not happen in our experiments. The most probable cause for this behavior is that our prior was not very informative. We tried to tune different values of Gaussian covariance Prior but due to computational incompetency could not run experiments on different values of the prior. Also, Bayesian Mean performed almost similar to the Bayesian over all possible posterior sample parameters. However, SVC performed better than Logistic Regression MAP and MLE.

6.4 Batch vs. Pool-based Uncertainty Sampling

6.4.1 Accuracy

Figure 5 shows that Batch Active learning accuracy is almost comparable. However, there is a slight improvement in the pool based active learning accuracy after a certain number of queries (70). We expected the pool based accuracy to be consistently higher but the results show that since the batch size was not too big (batch size = 10), the results are almost comparable.

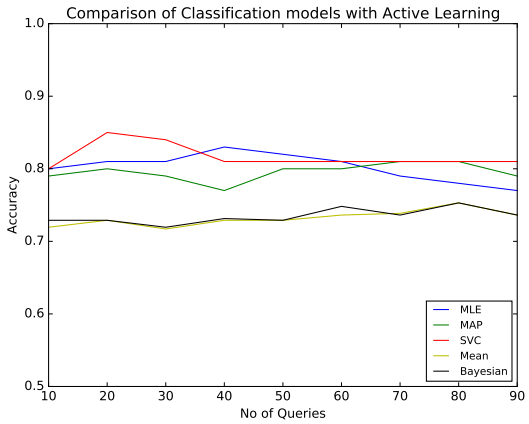

Figure 4: Comparing Active Learning on all classifiers

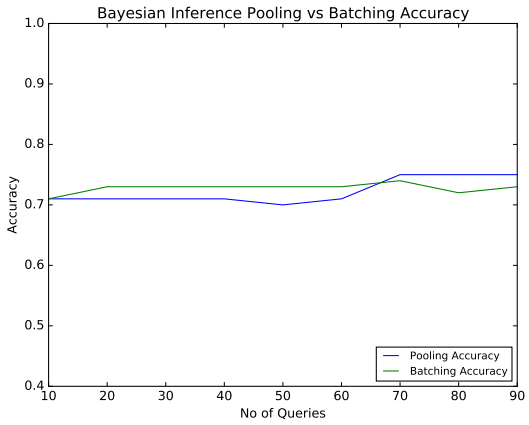

Figure 5: Comparing Accuracy of Batch vs Pool based Active Learning

6.4.2 Runtime

Figure 6 shows a clear picture of the exponential rise in pool active learning runtime as the number queries increases. However, batching almost takes a considerably constant amount of time for different number of query sizes. This is why we used batching in all other experiments involving Bayesian Active Learning.

6.5 Hyper-parameter Tuning

6.5.1 Regularization Tuning for Support Vector Classification

In Figure 7, we can see that a greater C value performs better than C=1. For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane mis-classifies more points. For comparing SVC Active Learning, we have hence chosen C=1 since we can get more data points in the soft margin and hence decide better on the most uncertain samples to label.

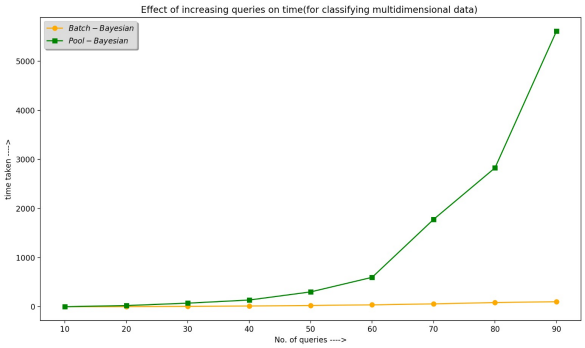

Figure 6: Comparing Runtime of Batch vs Pool based Active Learning

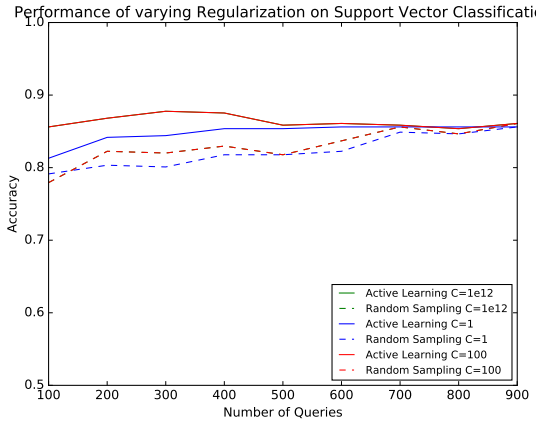

Figure 7: Hyper-parameter Tuning for SVC

6.5.2 Logistic Regression Tuning Regularization

As seen in figure 8, we find that larger the C value(regularization) better is the accuracy of the model for less data points. This is because when data size is small, the regularization helps in giving some weight to the outliers and hence performs better. For θ_{MLE} we hence selected a higher value of C so that the effect of regularization is null.

6.5.3 Bayesian Inference Tuning Gaussian Prior Covariance

Figure 9 shows us a comparison of the different prior covariances (5,10, 15, 20) on Bayesian Active Learning. As we can see that the results are almost comparable. This is because there is not much difference in the different covariance prior values. However, we could not run this experiment with higher values of covariance because with increase in prior value the runtime was increasing exponentially. We did not have enough computational capacity, hence we chose to run the experiment on smaller values of prior and visualize the effect of it on our prediction models.

6.5.4 Effect of increasing posterior sample size on Rejection Sampling

Increasing posterior size did not really improve our accuracy results as observed in Fig 10 (although they were comparable.). Also, the runtime involved with increasing sample posterior size increased beyond our computational scope, hus, rest of tests were fixed at posterior sample size = 100.

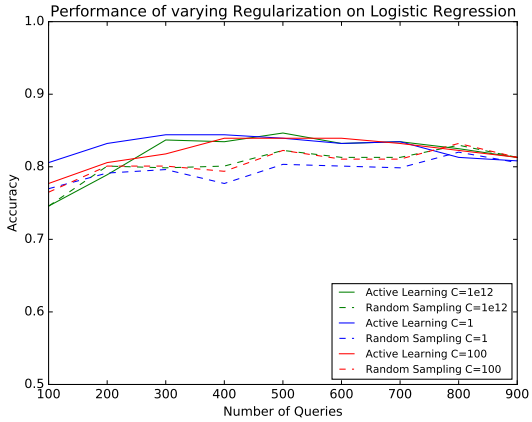

Figure 8: Hyper-parameter Tuning for Logistic Regression

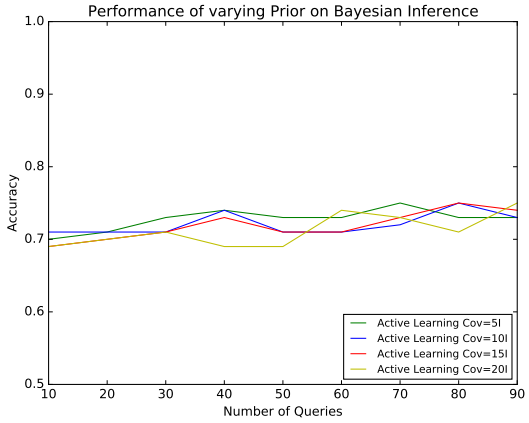

Figure 9: Tuning prior covariance for Bayesian Logistic Regression

6.5.5 Effect of Increasing Unlabeled sample size on Bayesian Active Learning

As seen in Figure 11, increasing the size of unlabeled dataset to choose the queries from did not make much of a difference on the accuracy curve. Since the rejection sampling is random in nature, the accuracies took a dip for a couple of instances, but otherwise was almost constant throughout.

7 Conclusion & Future Work

Bayesian Active Learning was throughout better performing than Bayesian Random Sampling. Logistic Regression Active Learning performed better than SVM Active Learning for less data as well as better than their Random Learning accuracies. Not enough data was the reason as to why both the models were fast but accuracy upto only 83% was achieved. When comparing with Bayesian over small data, SVC performed the best because 100 points of the multidimensional data were not enough for good performance of the batch-Bayesian model. We think that increasing the sample posterior value to a higher value would have resulted in better accuracy with the pool-based Bayesian Active Learning model but this was computationally intensive for evaluation purposes.

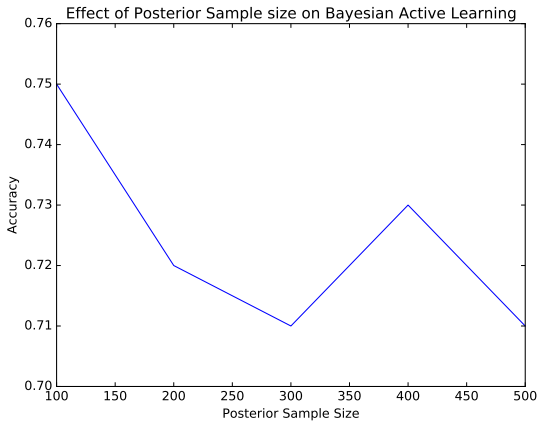

Figure 10: Tuning posterior sample size for Rejection Sampling

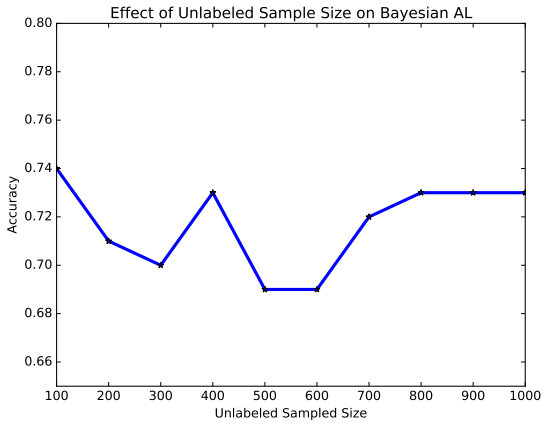

Figure 11: Effect of unlabeled sample size on Bayesian Active Learning

However, we could verify that batch-bayesian performs faster but poorer than pool-based bayesian model. Also, rejection sampling over all the posterior samples gave better active learning accuracies than the mean posterior .
In future, we plan to extend our study to multi-class classification problem. Better results can be observed for Bayesian Inference, θ_{Mean} plug-in estimate and θ_{MLE} plug-in estimate if we have a better informative prior. This study can hence be extended to optimizing the prior covariance parameter. Also, since the dataset is multidimensional, having better computational resources will help perform even more experiments and get promising results.

8 References

[1] Burr Settles, University of Wisconsin–Madison (2009) *Active Learning Literature Survey.*Computer Sciences Technical Report.

[2] X. Zhu. *Semi-Supervised Learning with Graphs*. PhD thesis, Carnegie Mellon University, 2005a.

[3] B. Settles, M. Craven, and L. Friedland. *Active learning with real annotation costs*. In Proceedings of the NIPS Workshop on Cost-Sensitive Learning, pages 110, 2008a.

[4] *Womens Health Risk Assessment*,https://gallery.cortanaintelligence.com/Competition/Womens-Health-Risk-Assessment-1

[5] *Dataset Description* . https://az754797.vo.msecnd.net/competition/whra/docs/data-description.docx

[6] Imputer Library.*Scikit-Learn Imputer*,http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.Imputer.html

[7] Schein et. al,*Active learning for logistic regression: an evaluation* Springer Science+Business Media, LLC 2007

[8] Tong et. alto , *Support vector machine active learning with applications to text classification*, Journal of machine learning research, 2001

[9] *Rejection Sampling,* https://en.wikipedia.org/wiki/Rejection_sampling

[10] Golovin et. al , *NearOptimal Bayesian Active Learning with Noisy Observations*, NIPS 2010

[11]*Steven C. H. Hoi,Rong Jin,Jianke Zhu,Michael R. Lyu, (2006) "Batch mode active learning and its application to medical image classification"*, doi 10.1145/1143844.1143897

[12]*Anbu et. al (2017) "Machine Learning Approach for Predicting Womens Health Risk"*, ICACCS -2017

[13]*Kremer et. al (2014) "Active Learning with Support Vector Machines"*, WIREs-Data Mining & Knowledge Discovery

[14]*Tong et. al "Active learning for structure in Bayesian networks"*, NIPS 2001

[15] *Roy et. al "Toward Optimal Active Learning through Monte Carlo Estimation of Error Reduction"*,ICML 2001