Journal of Statistical Research *https://doi.org/10.47302/jsr.2021550116*
2021, Vol. 55, No. 1, pp. 249-266 *ISSN 0256 - 422 X*

SEQUENTIAL AND ROBUST DATA SELECTION IN ACTIVE LEARNING FOR CLASSIFICATION

XIAOJIAN XU*

Department of Mathematics, Brock University
1812 Sir Isaac Brock Way, St. Catharines, ON, Canada L2S 3A1
Email: xxu@brocku.ca

CHARLIE SHAY

Sprout Studio, 110 James St, St. Catharines, ON, Canada, L2R 7E8
Email: garrett.shay@sproutstudio.com

SUMMARY

Active learning has become a popular learning process for classification. By selecting the most beneficial training data, an active classifier achieves better classification accuracy than a passive classifier. In this paper, we first investigate the methods of robustifying optimal active learning processes, via either a sequential approach or taking consideration of the classifiers possibly developed from a misspecified model. A comparison study has been presented for the classifiers obtained by a two-stage learning and a sequential learning as proposed and it indicates that the sequential method generally outperforms its competitor. Then, we further analyze the sensitivities of three different classifiers (linear discriminant classifier, quadratic discriminant classifier, and logistic regression classifier) in active learning for classification purpose. Our analysis reveals that the logistic regression classifier is sensitive to the misspecification involved in the assumed logistic model whereas the linear discriminant classifier is relatively robust to moderate violations of assumed homscedasticity.

Keywords and phrases: active learning; passive learning; robust design; logistic regression; Fisher's linear discriminant; heteroscedasticity.

AMS Classification: Primary 62K05, 62F35; secondary 62J12.

1 Introduction

In this paper, we discuss the methods of robustifying optimal active learning processes, via either a sequential approach or taking consideration of the classifiers possibly developed from a misspecified model. Recently, two-stage active learning methods have been developed in Xu and Shay (2020), where various classifers for classification and discrimination were discussed, including active linear discriminant classifier (ALD), active quadratic discriminant classifier (AQD), and active logistic

* Corresponding author

regression classifier (ALR). In the current paper, we first present a sequential active learning method. Then, we further analyze the sensitivities of these three different classifiers for active learning.

Although ALD and ALR have been commonly used techniques for classification, their applications, improved adaptations, and new developments specifically for active learning continue being investigated broadly in the literature. We name just a few recent ones: Yu et al. (2016), Yin et al. (2018), Hsu et al. (2019), as well as the references therein. One of the chief concerns in classification is the cost of obtaining true labels for observations. In many situations, the cost of labelling data can be quite high. Medical diagnoses, text classification, and speech recognition are such examples where financial cost or time cost becomes quite high (Settles, 2009). While passive learning can be used, they often require large random samples to provide good classification accuracies. Thus, one of the goals of active learning is to minimize the cost of training a classifier by selecting the most informative observations. The classifier first takes an initial set of labelled data, usually a small set. Then, it parses through the unlabeled set $\mathcal{U}$ to select an instance that can be labelled by a human annotator (often called an *oracle* in the literature). Once the point is labelled, it is added to the labelled set and the algorithm continues.

There exist several querying strategies to find the best training data. In this paper, we will focus on *pool-based sampling*. Pool-based sampling assumes that there is a large pool of unlabeled data from which the algorithm can make queries. In pool-based sampling, we can sample sequentially, evaluating instances one at a time and pulling an observation into the training set. Alternatively, we can evaluate all instances in $\mathcal{U}$ simultaneously, selecting the observations that the algorithm has evaluated as good training points. To make the decision about which instances should be queried, we will be using *uncertainty sampling*. Uncertainty sampling is a strategy for identifying unlabeled items that are near a decision boundary in the current machine learning. When an active learning queries the oracle, it must employ some criteria to decide which data in the unlabeled pool will benefit the training set the most. Uncertainty sampling is based on the idea of an algorithm selecting data that we are unsure how to classify. For probabilistic models, this becomes a very simple process. When classifying two groups, the algorithm queries the instances that have a posterior probability close to 0.5 (Lewis and Catlett, 1994; Lewis and Gale, 1994). With probabilistic models, the following approaches can be taken. The first is the *least confidence strategy*. The algorithm selects the instance, to query its actual label, in which has the least confidence in the most likely label. We denote such instance by $\boldsymbol{x}^*$. In this method, we have

$$\boldsymbol{x}^*_{LC} = \arg\max_{x \in \mathcal{U}} \Big(1 - P(\widehat{y} \mid \boldsymbol{x})\Big), \tag{1.1}$$

where

$$\widehat{y} = \arg\max_{y} P(y \mid \boldsymbol{x})$$

is the most probable label for instance $\boldsymbol{x}$**.** The second is *margin sampling* which takes into account the posterior of the second-most likely class. This algorithm selects the instance in which has the least difference between the two posterior probabilities of the most and the second-most likely labels. In this method, we have

$$\boldsymbol{x}^*_{M} = \arg\min_{x \in \mathcal{U}} \Big(P(\widehat{y_1} \mid \boldsymbol{x}) - P(\widehat{y_2} \mid \boldsymbol{x})\Big),$$

where $\widehat{y_1}$, and $\widehat{y_2}$ are the first and second most probable labels for instance $\boldsymbol{x}$. The third one is using *entropy* (Shannon, 1948), which is the most generalizable method for multi-class active learning. With entropy, we have

$$\boldsymbol{x}_H^* = \arg\max_{x \in \mathcal{U}} \Big(-\sum_i \big[P(y_i \mid \boldsymbol{x}) \log P(y_i \mid \boldsymbol{x}) \big] \Big)$$

with y_i ranging over all possible class labels. In the binary case, all three of these methods simplify to sampling the instance with the posterior probability closest to 0.5 (Settles, 2009). However, in the multi-class approach, entropy is often the most useful method for more complex tasks (Settles and Craven, 2008).

For non-probabilistic models, there has been some work on using uncertainty sampling with support vector machines (SVMs, Tong and Koller, 2001), which is similar in principle to probabilistic uncertainty sampling in the binary case. These methods sample the instances that are closest to the boundary line, however that boundary line is to be decided.

Previously, Xu and Shay (2020) looked at empirical results of probabilistic and non-probabilistic classifiers doing *two-stage* active learning (explained in detail in Section 2) in the *binary* case. Two-stage active learning requires that the classifier only seeks out the most beneficial training points once, after the first-stage random component. In this paper, we mainly consider the binary case of classification; however, we will examine a different approach: sequential active learning. We also briefly address the extension to the multi-class case of classification. In dealing with sequential active learning, we further adopt sub-sampling as a method of reducing computation time. In addition, we will discuss possible robustification of these forth-mentioned classifiers.

The remainder of this paper is organized as follows: Section 2 presents the sequential active learning process, including stopping criteria, computational time trade-off, and subsampling techniques that can be applied to improve the time efficiency. It also includes the model modifications as needed to extend our methods to multi-class active learning. Section 3 carries out an analysis for the sensitivities of ALD, AQD, and ALR classifiers as well as discusses the robustification methods against possible violations of model assumptions. Finally, we conclude this paper with a few remarks in Section 4.

2 Active Learning

When a *two-stage* active learning is adopted, the following two distinct stages are involved: (a) randomly sample some initial set of observations, and (b) use the information from the initial training to rank the remaining data points, picking the most beneficial ones simultaneously. This is a convenient and computationally efficient method of active learning, but it is heavily subject to the initial random sample. It has been shown that very small first-stage training sets can lead to poor classification accuracy and reduced performance. Therefore, we presently consider the process of *sequential active learning*.

Sequential active learning is simple in concept: rather than learning only once (which the two-stage method does), we can allow our algorithm to learn as each point is individually pulled into

the training set. The beginning part of the process is very similar to the two-stage method. We randomly sample the initial training set for the first stage, then we rank each point in the unlabeled set based on the criteria of the classifier, such as posterior probability or distance to the boundary. After the first stage, however, we diverge; rather than optimally selecting the top m points, where m is a prespecified number related to total training sample size as user-needed, we select only the best choice. That point is pulled into the training set, labelled, and the process repeats again by choosing the next best point.

The advantage to this method is that the classifier is constantly learning, and as it pulls more data into the training set, the classifier becomes better and better at distinguishing between classes. Two-stage active learning evaluates all actively selected points using only a first-stage passive training set; sequential active learning evaluates only its first point passively, and all subsequent points are evaluated using both the initial passive data and subsequent active data. Theoretically, as a result sequential active learning should perform more accurately than two-stage active learning. This will be demonstrated by our exemplary simulation in the next subsection.

2.1 Simulation

Ideally, an actively selected training set will achieve one of two goals, depending on the problem at hand: (a) require fewer labelled observations to achieve the same accuracy as a passively selected training set, or (b) achieve a higher accuracy than random sampling with the same number of labelled observations. Similar to Xu and Shay (2020), we seek to minimize a corresponding loss function set by picking the most beneficial observations. As previously mentioned, any probabilistic classifier is relatively simple to query. However, we can also use a non-probabilistic classifier, such as Fisher's linear discriminant. In this section, we will first adopt an ALD classifier, and an ALR classifier. With both approaches, we will select the training data by *uncertainty sampling.*

In general, for an ALD classifier, we assume the observed samples $\mathbf{x}_i^{(1)}$, $\mathbf{x}_j^{(2)}$ from two normal distributions with a common covariance. Namely,

$$\begin{aligned}\mathbf{x}_i^{(1)} &\sim \mathcal{N}_q(\mu_1,\ \mathbf{\Sigma}),\ i = 1, 2, \ldots, n_1, \text{ and}\\ \mathbf{x}_j^{(2)} &\sim \mathcal{N}_q(\mu_2,\ \mathbf{\Sigma}),\ j = 1, 2, \ldots, n_2.\end{aligned}$$

Then, the loss function for ALD training data selection can be defined as

$$l_1 = \left[(\mu_1 - \mu_2)^T \mathbf{\Sigma}^{-1} \left(\mathbf{x} - \frac{\mu_1 + \mu_2}{2}\right)\right]^2, \tag{2.1}$$

or

$$l_{11} = \sum_{i=1}^{m} \left[(\mu_1 - \mu_2)^T \mathbf{\Sigma}^{-1} \left(\mathbf{x}_i - \frac{\mu_1 + \mu_2}{2}\right)\right]^2, \tag{2.2}$$

where $\mathbf{x}$ or $\mathbf{x}_i$ is any possible candidate data point to be selected into a training sample. In the case of sequential selection method, the optimal training data can be sequentially selected by minimizing (2.1). In the case of two-stage selection method, the optimal training set of m data points can be selected at once by minimizing (2.2).

We may (adaptively) compute

$$\overline{\mathbf{x}}_1 = \frac{1}{n_1}\sum_{i=1}^{n_1}\mathbf{x}_i^{(1)}, \quad \overline{\mathbf{x}}_2 = \frac{1}{n_2}\sum_{j=1}^{n_2}\mathbf{x}_j^{(2)}, \tag{2.3}$$

and

$$\mathbf{S}_{\text{pooled}} = \frac{1}{n_1+n_2-2}\left[\sum_{i=1}^{n_1}\left(\mathbf{x}_i^{(1)}-\overline{\mathbf{x}}_1\right)\left(\mathbf{x}_i^{(1)}-\overline{\mathbf{x}}_1\right)^T + \sum_{j=1}^{n_2}\left(\mathbf{x}_j^{(2)}-\overline{\mathbf{x}}_2\right)\left(\mathbf{x}_j^{(2)}-\overline{\mathbf{x}}_2\right)^T\right]. \tag{2.4}$$

Then, the loss functions in (2.1) and (2.2) can be estimated as

$$\widehat{l_1} = \left[(\overline{\mathbf{x}}_1-\overline{\mathbf{x}}_2)^T\mathbf{S}_{\text{pooled}}^{-1}\left(\mathbf{x}-\frac{\overline{\mathbf{x}}_1+\overline{\mathbf{x}}_2}{2}\right)\right]^2, \tag{2.5}$$

and

$$\widehat{l_{11}} = \sum_{i=1}^{m}\left[(\overline{\mathbf{x}}_1-\overline{\mathbf{x}}_2)^T\mathbf{S}_{\text{pooled}}^{-1}\left(\mathbf{x}_i-\frac{\overline{\mathbf{x}}_1+\overline{\mathbf{x}}_2}{2}\right)\right]^2,$$

respectively.

For an ALR classifier, we estimate the probability of an observed subject coming from the first population, $p = P(Y=1)$, via a logistic regression. We assume

$$\ln\frac{p}{1-p} = \beta_0 + \boldsymbol{\beta}_1^T\mathbf{x}, \tag{2.6}$$

where $\mathbf{x}$ contains q explanatory variables. The loss function for ALR is

$$l_2 = \left(\frac{e^{\beta_0+\boldsymbol{\beta}_1^T\mathbf{x}}}{1+e^{\beta_0+\boldsymbol{\beta}_1^T\mathbf{x}}} - 0.5\right)^2, \tag{2.7}$$

or

$$l_{22} = \sum_{i=1}^{m}\left(\frac{e^{\beta_0+\boldsymbol{\beta}_1^T\mathbf{x}_i}}{1+e^{\beta_0+\boldsymbol{\beta}_1^T\mathbf{x}_i}} - 0.5\right)^2. \tag{2.8}$$

In the case of sequential selection method, the optimal training data points $\mathbf{x}$ can be sequentially selected by minimizing (2.7). In the case of two-stage selection method, the optimal training set of m data points $\mathbf{x}_i$ can be selected at once by minimizing (2.8). With labeled data provided, β_0 and $\boldsymbol{\beta}_1$ can be (adaptively) estimated via the assumed logistic regression model using maximum likelihood method, for instance.

For our simulations, we take $q = 2$ for simplicity. Supposingly, it is assumed to classify with two bivariate normal populations having a common correlation matrix. We set the parameters (two population means and correlation matrix) as

$$\mu_1^T = (0.5, 0),\ \mu_2^T = (-0.5, 0),\ \text{and } \mathbf{R}_1 = \mathbf{R}_2 = \begin{pmatrix} 1 & -0.5 \\ -0.5 & 1 \end{pmatrix}.$$

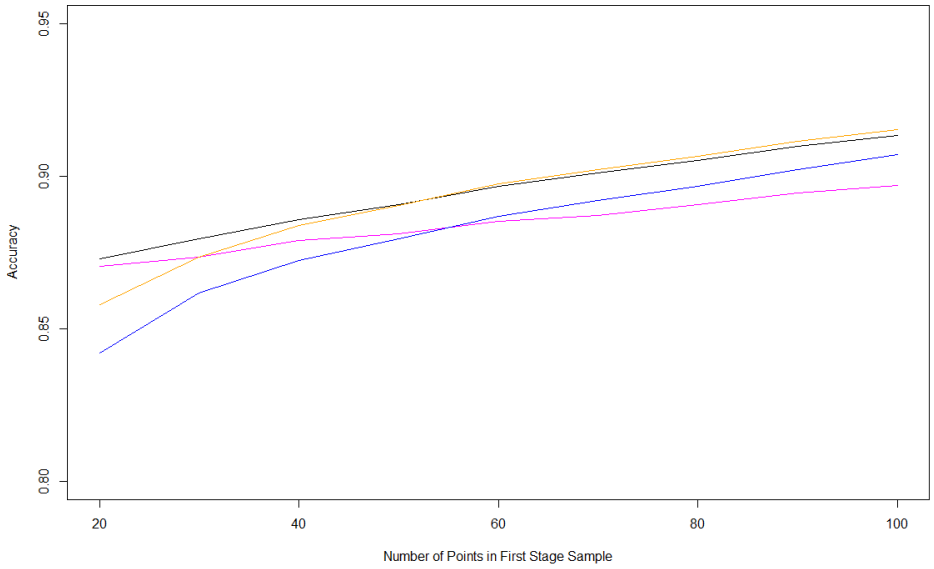

Figure 1: Accuracy as the actively selected set increases: sequential ALD (black), two-stage ALD (magenta), sequential ALR (orange), and two-stage ALR (blue).

In this simulation, we fix the first-stage sampling at $n_{\text{rand}} = 15$, varying only the active component of the algorithm. Additionally, we set the "unlabeled" pool size to 800.

We measure the performance of a classifier by the proportion of correctly classified data points among the 800 points in the "unlabeled" pool. We name it as "Accuracy" in our figures. For instance, Figure 1 presents the accuracy as the size of actively selected set increases for sequential and two-stage ALD or ALR classifiers. We can see that in general, sequential active learning outperforms two-stage active learning for the ALD; there is also a small but noticeable increase in computation time that occurs, which we will discuss in Section 2.3. The performance difference becomes more noticeable in extreme cases, such as in the case of first-stage sample size being very small. Our simulation results show impressive performance provided by the sequential ALD classifier.

Next, we fix the actively selected sample size to be $n_{\text{active}} = 35$ and vary n_{rand} from 4 to 20. With a small n_{rand}, the sequential ALD still performs very well, especially in comparison with its two-stage counterpart. There is thus an obvious benefit to a sequential ALD in both highly constrained training set sizes. It has also noticed that the sequential ALR is consistently outperform its two-stage competitor. The comparison of their performances for this typical example has been demonstrated by Figure 2.

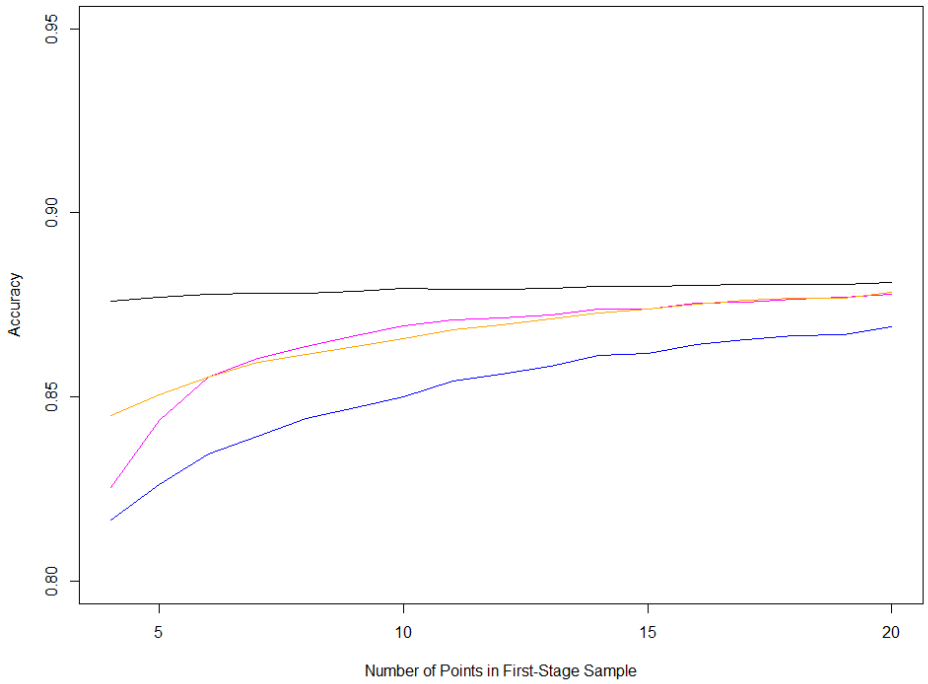

Figure 2: Accuracy of the four classifiers under small n: sequential ALD (black), two-stage ALD (magenta), sequential ALR (orange), and two-stage ALR (blue).

2.2 Stopping criteria

Another advantage of sequential active learning, as opposed to two-stage, is its ability to utilize a stopping criterion to stop the classifier from learning as soon as sufficient performance has been reached. There are a number of stopping criteria in the active learning literature which we briefly review here. Intuitively, an active learning classifier should stop learning when adding additional observations ceases to yield any meaningful change in performance: in other words, when the cost of annotating actively selected observations outweighs the benefits of adding those observations to the training set. There are issues with this interpretation; however, in order to assess the accuracy of a classifier, we would need some test set, properly annotated, so that we can make some judgement on the performance of the classifier. These tests are not always available and often expensive to get. Since one of the objectives of active learning is to achieve high performance with low cost, this is not an ideal scenario.

We are thus searching for some way of generating a stopping criterion based on some aspect of model performance outside of raw accuracy. There are several options. The first option focuses on

assessing the model's confidence in its ability to classify the unlabeled set. We can also implement a criterion which is confidence-based, where the learner will stop learning when the model's confidence drops (Vlachos, 2008). One issue with this stopping criterion is that it assumes the learner is incapable of fully explaining all examples; if this is not the case (i.e., it is capable of properly classifying its training set), the performance of the stopping criterion drops. Confidence can be estimated in a variety of ways. Vlachos (2008) defines confidence in the case of logistic regression as the average of the entropies in the test set, for example.

Another confidence method measures the gradient of confidence estimates (Laws and Schutze, 2008). In this method, a threshold of model confidence is pre-set by the user, and the rate of change of the model's confidence is assessed by examining recent points pulled into the training set. If the gradient dips below the threshold, the classifier stops learning. Essentially, this method seeks a point when the classifier's performance levels out. Since we have no true test set to compare to, we assess performance by confidence in labelling the unlabeled set, rather than by classification accuracy. In the gradient method, instead of stopping when the classifier reaches a sufficient confidence threshold, we stop instead when the confidence no longer increases. This method bears some similarity to Vlachos' method rather than focusing on hitting a confidence threshold. In this method, one seeks to stop when the rate of change of confidence hits an elbow point, namely when new observations yield adding less confidence to the model.

In place of stopping when confidence grows too low, we can also utilize a maximum confidence criterion. In this criterion, a threshold is pre-set by the user, and the learner stops learning when its confidence in each unlabeled example exceeds the pre-established threshold. Thus, the learner has requisite confidence in its ability to classify, such that additional training points are unnecessary. We note that the maximum confidence is not the same as Vlachos'. Whereas Vlachos', as well as that of Laws and Schutze (2008), establish confidence as an estimate covering the entirety of the unlabeled pool, maximum confidence is a threshold that must be met by each observation in the unlabeled set.

There are also stopping criteria that are not confidence based. One method is exclusive to margin-based classifiers such as SVMs, and is thus named margin-exhaustion. In this criterion, the learner stops learning when all unlabeled examples fall outside the classifier's margin. There are two problems unique to this stopping criterion: (i) this method cannot extend to probabilistic classifiers, and (ii) some empirical work has demonstrated that margin exhaustion tends to stop late, and it often continues to learn beyond the point of benefit (Schohn and Cohn, 2000).

We can also attempt a minimum-error stopping criterion, which avoids the problems covered in the beginning of this subsection. Rather than assessing accuracy on an external test set, the learner can stop learning once its accuracy on the queried training set reaches some threshold of accuracy (Zhu and Hovy, 2007). We thus avoid the issue of annotating a test set, which can be expensive, while still maintaining a performance-based classifier that is a very intuitive approach to stopping criteria.

Deciding which stopping criterion to use is subject to the goals of the classifier as well as the goals of the investigator. Settles (2009) notes that choice of stopping criteria, if one is even employed, is usually problem-contextual and constrained by non-theoretical factors. For instance, if the goal is to achieve the best accuracy with a constrained sample size, there are likely financial

Table 1: Runtime increased compared to two-stage ALD

Sequential ALD	Sequential ALR	Two-stage ALR
1057%	13829%	357%

costs that are more pertinent than achieving some threshold. The other main issue with implementing these criteria is their reliance on pre-established thresholds. A prior knowledge of the best threshold is likely rare, and while general thresholds can be implemented, there is no guarantee that those thresholds are optimal to achieve the best performance-to-cost ratio.

As mentioned above, adding in the computation of stopping criteria will elongate the computation time of the classifier, as the stopping criteria will need to be evaluated at each point's annotation, determining whether more points are needed.

2.3 Subsampling

While we have seen the distinct advantages to using sequential active learning over two-stage active learning, there is a demerit as well. The disadvantage of using sequential learning is that of computation cost. The cost of evaluating all instances in an unlabeled set simultaneously is relatively small; in our simulations, a two-stage ALD even for an unlabeled pool around 8 million took only a few seconds. Thus, implementing two-stage active learning is time-efficient to perform. However, sequential active learning will evaluate the entire unlabeled set for each actively selected point added to the set. If we consider also evaluating a stopping criterion at each point of the algorithm, we can see that the computation time of the algorithm increases heavily as the unlabeled set grows. Given that many datasets are incredibly large, with ostensibly millions of observations, the issue of computation time is an extremely relevant one. The simulation runtime was recorded with $n_{\text{rand}} = 15$ and $n_{\text{active}} = 35$, and by setting the unlabeled set as large as $n_1 = n_2 = 4,000,000$. For this example, the runtime for two-stage ALD, under our simulation condition, was only 0.07 second. Table 1 provides the example runtime increased by percentage relative to their competitor two-stage ALD under the same conditions. It is still preferable to use the sequential active learners, as it is clear they have advantages of accuracy gains, but there is an accuracy/runtime trade-off. As the number of predictors increases, the issue of high dimensionality also arises and affects computation time. With so many factors contributing to the computation time for sequential active learning, it is clearly of interest to reduce the computational time.

There are ways of overcoming or mitigating this issue. One approach is a compromise between purely two-stage or purely sequential active learning, which we could call a micro-two-stage approach. Rather than evaluating the unlabeled set for each actively selected instance, one can instead evaluate the unlabeled set for each actively selected group of instances. The choice of two-stage size can be sought after using a variety of methods (Settles, 2009). This certainly mitigates the computation cost problem, but it does not solve it entirely.

Another method to aid in reducing the cost is sub-sampling, selecting some subset of the entire

unlabeled set with whom we can build our model, and that is what we will discuss next.

Our simulation results have shown that computation time for a sequential learner increases both as the training set increases and as the unlabeled set increases, especially for the ALR case. It is of interest, then, to minimize computation time while keeping accuracy high. We thus propose subsampling as a viable method of reducing computation time.

Optimal subsampling (sampling the most beneficial observations to build a model upon) is a very similar process to active learning. The idea behind optimal subsampling is to pick the 'best' possible observations in the sample to build the model. Here, 'best' can be defined in various ways. Generally, optimal sampling designs seek to minimize the variances, of the estimation or prediction for a specified quantity of interest, by maximizing the amount of information in the chosen observations. Such maximization is often performed based on the information matrix, which often subsequently minimizes a scalar function of the covariance matrix (Smith, 1918).

Subsampling is also particularly relevant for so-called big data problems: data sets which are incredibly large and are thus computationally inefficient with which to build a model. Wang et al. (2018) have recently proposed an optimal subsampling design for large-sample logistic regression, which assigns optimal subsampling probabilities (SSP) to minimize the variance of the maximum likelihood estimator obtained from sample data of size n. The optimal SSP π_i takes the form

$$\pi_i = \frac{\left| y_i - p_i\left(\widehat{\boldsymbol{\beta}}_{MLE}\right) \right| \|\mathbf{x}_i\|}{\Sigma_{j=1}^{n} \left| y_j - p_j\left(\widehat{\boldsymbol{\beta}}_{MLE}\right) \right| \|\mathbf{x}_j\|}, \; i = 1, 2, \ldots, n,$$

where $\widehat{\boldsymbol{\beta}}_{MLE}$ is the maximum likelihood estimator of $\boldsymbol{\beta} = \left(\beta_0, \; \boldsymbol{\beta}_1^T\right)^T$, and $p_i\left(\widehat{\boldsymbol{\beta}}_{MLE}\right)$ is the posterior probability function,$(e^{\beta_0 + \boldsymbol{\beta}_1^T \mathbf{x}})/(1 + e^{\beta_0 + \boldsymbol{\beta}_1^T \mathbf{x}})$ evaluated at $\widehat{\boldsymbol{\beta}}_{MLE}$ and $\mathbf{x}_i$. Since the optimal SSP depends on $\boldsymbol{\beta}$, it must be initially estimated. Therefore, as the first step, a non-optimal SSP, such as uniform, is often used to obtain an initial estimate. Then, the second step of the algorithm is performed using the optimal SSP to subsample the best possible observations.

This optimal subsampling algorithm is efficient and computationally reasonable; however, because the algorithm relies on the true response value y_i, this optimal algorithm is not available for this particular problem. Some future work will be done in this direction. Fortunately, we can instead assign uniform subsampling probabilities, which is equivalent to randomly sampling from the dataset. Because we are still choosing the most beneficial observations, the potential lack of optimality of the subsample is less of an issue. Figure 3 shows the results of a simulation study investigating the accuracy under random subsampling. We can see that a fixed random subsample allows consistent accuracy gains across unlabeled pool size.

2.4 Sequential active learning for multiple classes

The methods discussed above can be extended to an active learning process for classification that involves more than two classes. In the case of k-classes, binary logistic regression can be replaced by its counterpart, multinomial logistic regression. The model can be expressed, by the probability

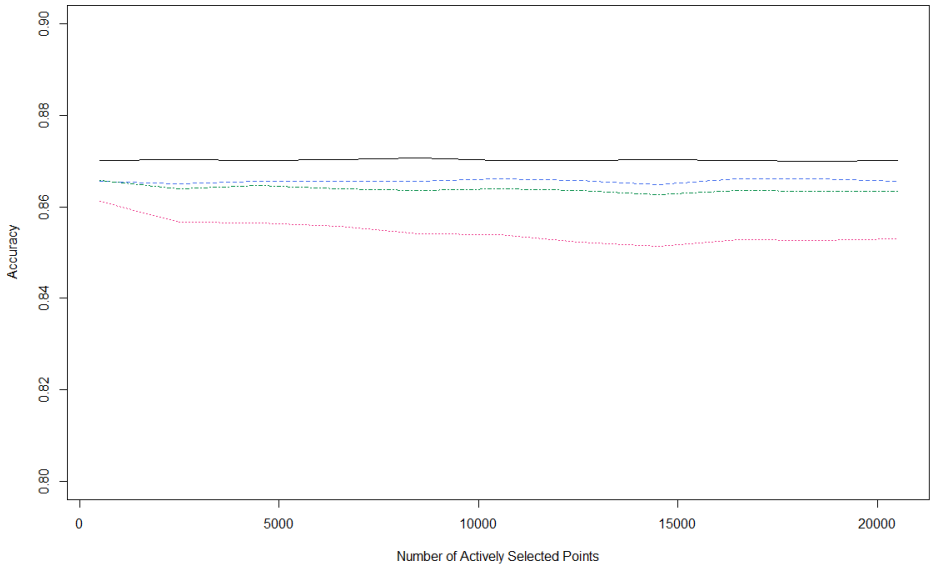

Figure 3: Accuracies of sub-sampled sequential active learning compared to full-sample two-stage learning: sequential ALD (black), two-stage ALD (blue), sequential ALR (green), and two-stage ALR (magenta).

of an instance $\mathbf{x}$ being a j^{th} class item, as follows

$$\Pr(Y=j) = \begin{cases} \frac{e^{\beta_{0j}+\boldsymbol{\beta}_j^T\mathbf{x}}}{1+\sum_{c=1}^{k-1} e^{\beta_{0c}+\boldsymbol{\beta}_c^T\mathbf{x}}}, & j=1,\ldots,k-1; \\ \frac{1}{1+\sum_{c=1}^{k-1} e^{\beta_{0c}+\boldsymbol{\beta}_c^T\mathbf{x}}}, & j=k. \end{cases}$$

Then, an observation can be labeled as j if such j provides the maximum probability. In this case, we redefine the loss function (2.7) to be

$$l_2 = \sum_{j=1}^{k-1}\left(\frac{e^{\beta_{0j}+\boldsymbol{\beta}_j^T\mathbf{x}}}{1+\sum_{c=1}^{k-1} e^{\beta_{0c}+\boldsymbol{\beta}_c^T\mathbf{x}}} - \frac{1}{k}\right)^2 + \left(\frac{1}{1+\sum_{c=1}^{k-1} e^{\beta_{0c}+\boldsymbol{\beta}_c^T\mathbf{x}}} - \frac{1}{k}\right)^2.$$

In order to adapt multinomial logistic regression for active learning, one simple option can be the *least confidence* method as defined in (1.1); that is, we select the observations with low posterior probabilities for their most likely class. These observations are, as the name implies, the ones about which the learner is least confident. We can thus construct an active multinomial regression classifier that sequentially selects the instance with the least confidence in the most likely class.

For a multi-class linear discriminant analysis, our proposed sequential ALD can be modified by

selecting observations optimally so as to minimize a multi-class version of (2.5) as follows

$$\widehat{l}_1 = \sum_{1\le i<j\le k} \left[(\overline{\mathbf{x}}_i - \overline{\mathbf{x}}_j)^T \mathbf{S}_{\text{pooled}}^{-1} \left(\mathbf{x} - \frac{\overline{\mathbf{x}}_i + \overline{\mathbf{x}}_j}{2} \right) \right]^2,$$

where $\overline{\mathbf{x}}_j = n_j^{-1} \sum_{i=1}^{n_j} \mathbf{x}_i^{(j)},\ j = 1, \ldots, k$, and

$$\mathbf{S}_{\text{pooled}} = \frac{1}{\sum_{j=1}^k n_j - k} \left[\sum_{j=1}^k \sum_{i=1}^{n_j} \left(\mathbf{x}_i^{(j)} - \overline{\mathbf{x}}_j \right) \left(\mathbf{x}_i^{(j)} - \overline{\mathbf{x}}_j \right)^T \right].$$

To create an optimal active training set, the multi-class active discriminant classifier selects the observation that has the least distance to their class boundaries.

3 Sensitivity and Robustification

It is worth examining the assumptions of our classifiers, and their sensitivity to those assumptions being violated. We will start with the assumptions of the linear discriminant analysis classifier, and then move on to model misspecification in the case of the logistic regression classifier.

3.1 Quadratic discriminant analysis

For an ALD classifier, we previously held an assumption of equality of covariances. If this assumption is not held, then we have the following:

$$\begin{aligned} \mathbf{x}_i^{(1)} &\sim \mathcal{N}_q(\mu_1,\ \mathbf{\Sigma}_1),\ \ i = 1, 2, \ldots, n_1, \text{ and} \\ \mathbf{x}_j^{(2)} &\sim \mathcal{N}_q(\mu_2, \mathbf{\Sigma}_2),\ \ j = 1, 2, \ldots, n_2. \end{aligned}$$

However, $\mathbf{\Sigma}_1 \neq \mathbf{\Sigma}_2$. With this modification, we accordingly define a new loss function to be minimized by the choice of $\mathbf{x}$:

$$\begin{aligned} l_q = \Bigg[&\frac{1}{2} \mathbf{x}^T (\mathbf{\Sigma}_1^{-1} - \mathbf{\Sigma}_2^{-1}) \mathbf{x} - \left(\mu_1^T \mathbf{\Sigma}_1^{-1} - \mu_2^T \mathbf{\Sigma}_2^{-1} \right) \mathbf{x} + \\ &\frac{1}{2} \left(\mu_1^T \mathbf{\Sigma}_1^{-1} \mu_1 - \mu_2^T \mathbf{\Sigma}_2^{-1} \mu_2 + \ln \frac{|\Sigma_1|}{|\Sigma_2|} \right) \Bigg]^2. \end{aligned} \tag{3.1}$$

We may (adaptively) compute $\overline{\mathbf{x}}_1, \overline{\mathbf{x}}_2$ by (2.3) and

$$\mathbf{S}_1 = \frac{1}{n_1 - 1} \sum_{i=1}^{n_1} \left(\mathbf{x}_i^{(1)} - \overline{\mathbf{x}}_1 \right) \left(\mathbf{x}_i^{(1)} - \overline{\mathbf{x}}_1 \right)^T,\ \mathbf{S}_2 = \frac{1}{n_2 - 1} \sum_{j=1}^{n_2} \left(\mathbf{x}_j^{(2)} - \overline{\mathbf{x}}_2 \right) \left(\mathbf{x}_j^{(2)} - \overline{\mathbf{x}}_2 \right)^T.$$

Then, the loss functions in (3.1) can be estimated as

$$\widehat{l}_q = \left[\frac{1}{2} \mathbf{x}^T (\mathbf{S}_1^{-1} - \mathbf{S}_2^{-1}) \mathbf{x} - \left(\overline{\mathbf{x}}_1^T \mathbf{S}_1^{-1} - \overline{\mathbf{x}}_2^T \mathbf{S}_2^{-1} \right) \mathbf{x} + \frac{1}{2} \left(\overline{\mathbf{x}}_1^T \mathbf{S}_1^{-1} \overline{\mathbf{x}}_1 - \overline{\mathbf{x}}_2^T \mathbf{S}_2^{-1} \overline{\mathbf{x}}_2 + \ln \frac{|\mathbf{S}_1|}{|\mathbf{S}_2|} \right) \right]^2.$$

Because the heteroscedasticity introduces a quadratic term, this classifier is preferably used for *quadratic discriminant analysis*, and we name our active version as the active quadratic discriminant classifier (AQD) as previously mentioned.

On the other hand, the sensitivity of the ALD to violation of the assumption of homoscedasticity is of interest. For this test, data is generated according to the following parameters: $\mu_1^T = (t, 0)$ and $\mu_2^T = (-t, 0)$ with initial $t = 0.5$, $\mathbf{\Sigma}_1 = \left(\begin{smallmatrix} 1 & -.707 \\ -0.707 & 2 \end{smallmatrix}\right)$, and $\mathbf{\Sigma}_2 = \left(\begin{smallmatrix} 1 & \sigma_{12} \\ \sigma_{21} & \sigma_{22}^2 \end{smallmatrix}\right)$ with the variance σ_{22}^2 ranging from .1 to 7.1 and $\sigma_{12} = \sigma_{21} = \pm 0.5\sqrt{\sigma_{22}^2}$, thus holding a correlation matrix of $\mathbf{R}_2 = \left(\begin{smallmatrix} 1 & \pm 0.5 \\ \pm 0.5 & 1 \end{smallmatrix}\right)$. Figure 4 shows the accuracy comparison results between AQD and ALD by varying σ_{22}^2.

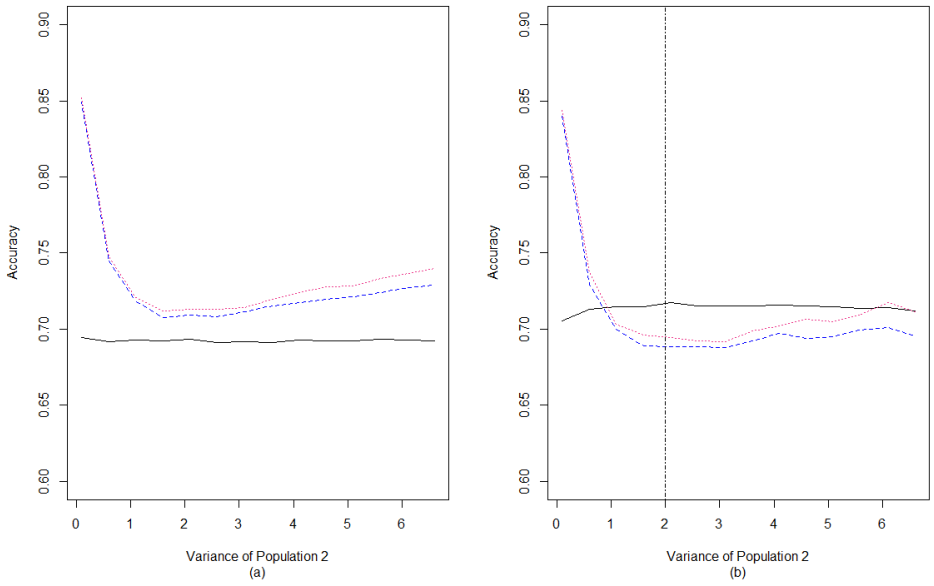

Figure 4: (a) Accuracy of the AQD vs ALD classifiers under growing positive covarinace and the second variance, and (b) accuracy under growing negative covariance and increasing second variance, with the point of perfect homoscedasticity at the vertical line. Here, we compare the accuracies among two-stage ALD (solid black), two-stage AQD (dash blue), and sequential AQD (dotted magenta).

We see that when the assumption of homoscedasticity is violated, the AQD tends to perform better than the ALD. As the covariance matrix $\mathbf{\Sigma}_2$ "shrinks", the AQD performs substantially better than the ALD. Also visible is the insensitivity of the linear discriminant method to violations of homoscedasticity; our results have shown that the performance of ALD remains relatively constant even as the inequality of covariance grows larger.

The parameters we have set result in a fairly difficult classification problem. We repeat the above simulation, but this time consider $t = 1$, holding the same parameters otherwise. This time we try

to test the performance of the AQD when the two populations are further apart and classification problem is easier to solve. Figure 5 presents the results of this new simulation. It has been shown that even under comparatively heteroscedastic conditions, when the problem becomes easier to solve, the performance of the AQD relative to the ALD diminishes.

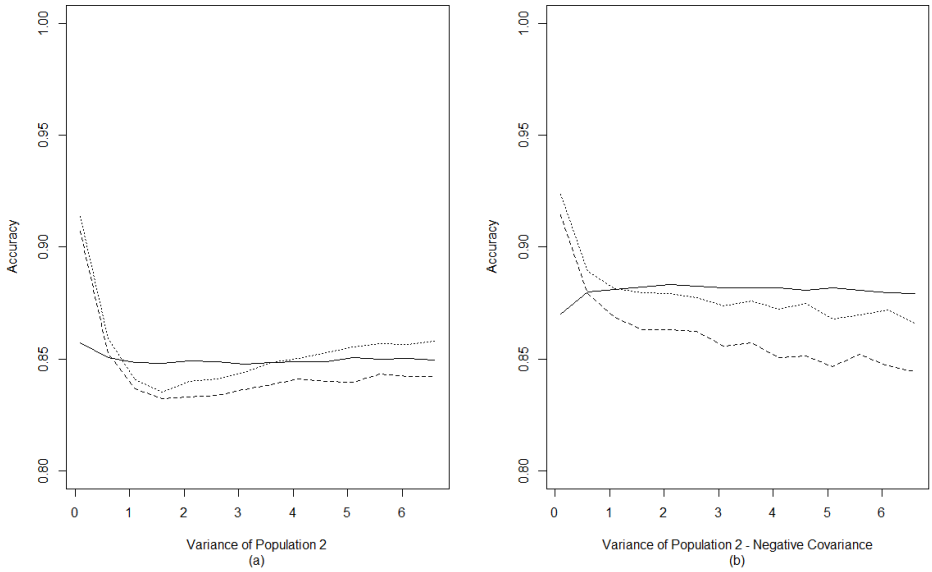

Figure 5: (a) Accuracy of the AQD vs ALD under growing positive covariance and variance, and (b) accuracy under growing negative covariance and increasing variance. Two-stage ALD (solid), two-stage AQD (dash) and sequential AQD (dotted).

3.2 Model misspecification for logistic regression

This subsection addresses how the ALR performs when the assumed model is misspecified. For the sake of computation time, we examine two-stage ALR, since the interest is not necessarily in comparing method accuracy, but sensitivity to misspecification. For the sensitivity analysis, we define the following performance measure

$$Se(\mathbf{x}) = \left(\frac{Am - At}{At}\right) \times 100\%, \tag{3.2}$$

where Am is the accuracy when the assumed model being misspecified, and At is the accuracy when the assumed model is correct. For this simulation, data are generated from two 3-variate populations with $\mu_1^T = (.5, .2, .4)$ and $\mu_2^T = (-.5, .2, -.4)$, with a common covariance. For demonstration

purpose, we take two different correlation structures: (i) the simplest correlation structure assuming

$$\mathbf{R}_1 = \mathbf{R}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \text{and (ii) } \mathbf{R}_1 = \mathbf{R}_2 = \begin{pmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 1 & 0.5 \\ -0.5 & 0.5 & 1 \end{pmatrix}.$$

We suppose that the actually true model is as specified as in (2.6) with $\boldsymbol{\beta}_1^T\mathbf{x} = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$ for (i), and $\boldsymbol{\beta}_1^T\mathbf{x} = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_1 x_2 + \beta_5 x_1 x_3 + \beta_6 x_2 x_3$ for (ii). However, an assumed model tends to be used for developing an ALR classifier. For both (i) and (ii), we consider two different misspecification that possibly occurred in an assumed model: (a) only two explanatory variables are involved, and the linear predictor is taking the form of $\beta_0 + \beta_1 x_1 + \beta_2 x_2$ in the fitted model, and (b) also only two explanatory variables are assumed and the linear predictor is taking into accounts of interaction this time, i.e., $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$. Figures 6 and 7 provide the results of the sensitivity analysis for Cases (i) and (ii) respectively, using the performance measure in percentage as defined in (3.2). This analysis reveals that the performance of the misspecified models grows weaker as more points are actively selected, also indicating that the ALR is very sensitive to model misspecification. In addition, when the true model's regressors are moderately correlated, the sensitivity to misspecification grows. Clearly then, robustifying ALR against model misspecification is a worthwhile endeavor.

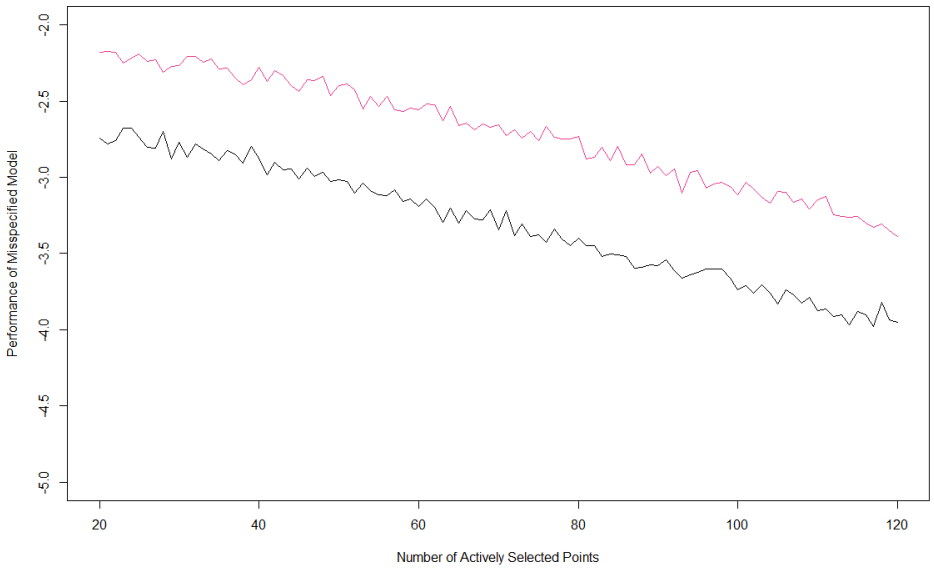

Figure 6: Reduced performance of an ALR classifier for misspecified models in percentage for Case (i). The black line is for misspecification of Type (a), and the magenta line is for that of Type (b).

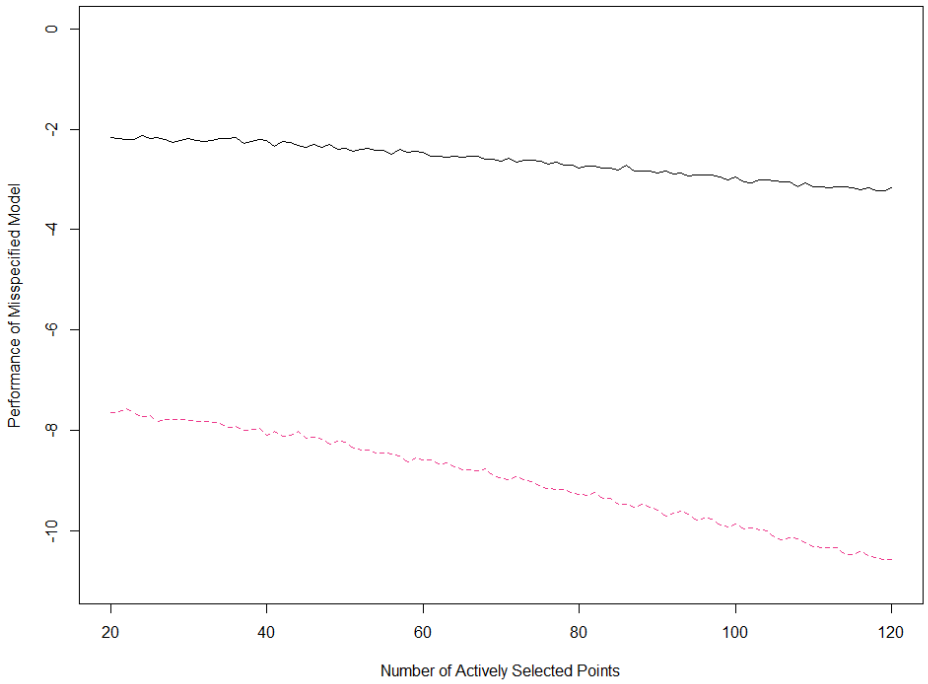

Figure 7: Reduced performance of an ALR classifier for misspecified models in percentage for Case (i). The black line is for misspecification of Type (a), and the magenta line is for that of Type (b).

Through our study, the model-based probabilistic classifiers are generally sensitive to the model assumptions. For desirable performance, they also require more first-stage samples compared to non-probabilistic classifiers. Therefore, we suggest the probability classifiers being adopted when there is sufficient training data in hand and the practitioner has confidence on the model being used. Otherwise, a non-probability classifier, especially at the first stage, is recommended.

4 Concluding Remarks

One of the interesting findings led by the results of this paper is the trade-off between using a linear discriminant and a quadratic discriminant. From our results, we can see that a violation of homoscedasticity does not necessarily guarantee that the AQD approach would perform better. In the situation of a classification problem involving more overlap among groups, the AQD seems to outperform the ALD, but in a less overlap involved situation, despite inequality of covariance, the ALD performs generally well. Adding to a practical issue, the AQD would be a better approach when

the classification "fairness" rather than overall accuracy is emphasized in a classification problem, and the AQD method should be a better choice when non-equality of covariances can be tested or suspected. If conducting a hypothesis test to assess equality of covariance is problematic, a natural question is to ask: when should one use ALD versus AQD for an active learning problem? This issue is especially pertinent because an actively selected training set comes from a biased distribution and is inherently tied to the classifier that selected it (Settles, 2009). Thus, constructing a training set using one classifier and then training on a separate classifier is not guaranteed to yield good results. Moreover, generally active learning can be conducted more efficiently when some knowledge of the underlying model or distribution is available, based on prior experimentation or model building. Diving into active learning algorithms with no understanding of the data is likely to lead to poor classifier performance. Lastly, our results have indicated that the ALD is more robust to violations of assumed covariance structures than others, and thus we would recommend it as the first step.

Acknowledgements

The research of both authors is supported by the Natural Sciences and Engineering Research Council of Canada.

References

Hsu, H., Chang, Y, and Chen, R. (2019), "Greedy active learning algorithm for logistic regression models," *Computational Statistics and Data Analysis*, 129, 119-134.

Laws, F., and Schutze, H. (2008), "Stopping criteria for active learning of named entity recognition," *Proceedings of the 22nd International Conference on Computational Linguistics,* 1, 465-472.

Lewis, D. D., and Catlett, J.(1994), "Heterogeneous uncertainty sampling for supervised learning," *Proceedings of the International Conference on Machine Learning (ICML)*, 148–156. Morgan Kauf-mann.

Lewis, D. D., and Gale, W. A. (1994), "A sequential algorithm for training text classifiers," *SIGIR'94*, 3-12. Springer, London.

Schohn, G., and Cohn, D. (2000), "Less is more: active learning with support vector machines," *ICML,* 2(4), 6.

Settles, B. (2009), *Active Learning Literature Survey*, Computer Sciences Technical Report 1648, University of Wisconsin-Madison.

Settles, B., and Craven, M. (2008), "An analysis of active learning strategies for sequence labeling tasks," *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1070-1079. Association for Computational Linguistics.

Shannon, C. E. (1948), "A mathematical theory of communication," *Bell System Technical Journal,* 27(3), 379-423.

Smith, K. (1918), "On the standard deviations of adjusted and interpolated values of an observed polynomial function and its constants and the guidance they give towards a proper choice of the distribution of observations," *Biometrika,* 12(1/2), 1-85.

Tong, S., and Koller, D. (2001), "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, 2, 45-66.

Vlachos, A. (2008), "A stopping criterion for active learning," *Computer Speech and Language,* 22(3), 295-312.

Wang, H. Y., Zhu, R., and Ma, P. (2018), "Optimal subsampling for large sample logistic regression," *Journal of the American Statistical Association.* 113(522), 829-844.

Xu, X. and Shay, C. (2020), "Optimal training data selection in active learning for discrimination and classification," *Proceedings of the 5th International Conference on Computer and Communication Systems (ICCCS)*, 15-19.

Yin, L., Wang, H., Fan, W., Zhou, J., and Yu, G. (2018), "Combining active learning and Fisher discriminant analysis for the semi-supervised process monitoring," *IFAC PapersOnLine* 15-21, 147-151.

Yu, X., Zhou, Y., and Ren, C. (2016), "An active learning based LDA algorithm for large-scale data classification," *International Journal of Databased Theory and Application*, 9(11), 29-36.

Zhu, J., and Hovy, E. (2007), "Active learning for word sense disambiguation with methods for addressing the class imbalance problem," *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning,* 783-790.

Received: February 2, 2021

Accepted: April 7, 2021