Master's Thesis

Sample reusability in importance-weighted active learning

Gijs van Tulder

Thesis defence: 31 October 2012
Student number: 1339389

MSc Media and Knowledge Engineering
Faculty Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Preface

Active learning is a simple but intriguing idea. It has a catchy name – *active* learning, who could be against that? – and the principle is so obvious that you wonder why it is not used more. Select the examples that give the most information to your classifier, so you can learn more with less – why not?

I first discovered active learning during the Machine Learning course, where I gave a presentation and wrote a short paper about the topic. Since then I found out that active learning is anything but simple: you have to be really careful if you want good results.

My thesis project focused on the dangers of active learning. In my literature survey I discussed the sample selection bias in active learning: because active learning does its best to make an unrepresentative sample selection, many conventional machine learning methods no longer work.

In the second phase of my project I explored the question of sample reusability, a result of the sample selection bias. Active learning selects the examples that a classifier finds interesting. This improves the performance of this specific classifier – it needs fewer samples to build a good model – but it is not clear if the selected examples are also useful to a *different* classifier. You do not want to use active learning if the other classifier learns more from a random set of samples than from the active sample selection.

Importance-weighted active learning is a recent active learning method that solves some of the problems of earlier active learning strategies. In this thesis I discuss whether it also solves the problem of sample reusability – it does not – and why reusability is a problem that is so hard to solve.

This thesis consists of two parts. The first part is my thesis paper, titled "Sample reusability in importance-weighted active learning". This is probably the part you would want to read. The second part is my "working document": a loose collection of unpolished thoughts, experiments and other material that I produced during my research. It shows some of the sidelines that I did not include in the paper.

I would like to thank Marco Loog, my thesis supervisor, for his advice, the interesting discussions and many cups of tea.

This thesis will be defended on 31 October 2012. The members of the thesis committee are prof. dr. ir. Marcel Reinders, dr. Charl Botha, dr. ir. Dick de Ridder, Veronika Cheplygina and my supervisor, dr. Marco Loog.

Gijs van Tulder
22 October 2012

Thesis paper

Sample reusability in importance-weighted active learning

Gijs van Tulder

22 October 2012

Recent advances in importance-weighted active learning solve many of the problems of traditional active learning strategies. But does importance-weighted active learning also produce a *reusable* sample selection? This paper explains why reusability can be a problem, how importance-weighted active learning removes some of the barriers to reusability and which obstacles still remain. With theoretical arguments and practical demonstrations, this paper argues that universal reusability is impossible: because every active learning strategy must undersample some areas of the sample space, classifiers that depend on the samples in those areas will learn more from a random sample selection. This paper describes several reusability experiments with importance-weighted active learning that show the impact of the reusability problem in practice. The experiments confirm that universal reusability does not exist, although in some cases – on some datasets and with some pairs of classifiers – there is sample reusability. This paper explores the conditions that could guarantee the reusability between two classifiers.

1 Introduction

Active learning is useful if collecting unlabelled examples is cheap but labelling those examples is expensive. An active learning algorithm looks at a large number of unlabelled examples and asks an oracle to provide the labels for the examples that look interesting. The premise is that labelling is expensive – it may involve asking a human or doing an experiment – so a good active learner will try to reduce the number of labels it requests; it will only query those examples that it expects will lead to the largest improvements of the model. If the active learner can do this it can achieve a lower *label complexity* than random sampling: by skipping useless examples, active learning can learn a better model with fewer labels.

Choosing the best examples is difficult: the learner does not know the labels in advance, so it must estimate how much each example might improve the model. Most algorithms base these estimates on the examples labelled in previous iterations (Figure 1). Some algorithms will also look at the distribution of the unlabelled samples. In general, almost every active learning strategy uses knowledge about the model that needs to be built: a different classifier may require a different sample selection.

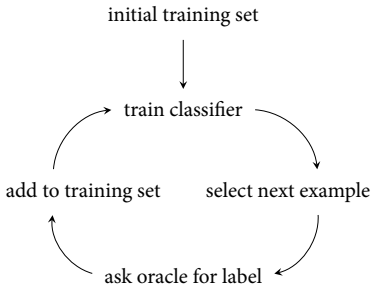


Figure 1: Active learning is an iterative process; most algorithms use the samples selected in a previous iteration to decide on the next sample.

For example, uncertainty sampling – one of the early active learning strategies, introduced by Lewis and Gale (1994) and still very popular – is a simple strategy that trains an intermediate classifier in each iteration and selects the example about which this classifier is most uncertain. In probabilistic classifiers the uncertainty is easily measured by the probability: if the classifier assigns a probability of 0.9 or 0.1 it is more certain of the label than if it assigns a probability close to 0.5. In support vector machines the uncertainty can be defined as the distance to the decision boundary: the sample with the smallest margin is the example that should be labelled next. Uncertainty sampling is simple to understand and easy to implement, but it is not without faults. Some of these faults will be discussed later.

Active learning works in two phases (Figure 2): first the selection strategy selects and labels examples – perhaps training some intermediate classifiers to aid in the selection – and produces a set of labelled examples. The classifier that is used in this phase is the *selector*. In the second step, the collected examples are used to train a final classifier that can be used to classify new examples. This classifier is the *consumer*.

Most active learning methods assume that the selector and the consumer are the same. This scenario is *self-selection*: the classifier algorithm that is used during the selection is also used to train the final classifier. Self-selection gives the best chances for active learning, since the best way to learn the value of a new example for a classifier is to ask the classifier itself. Perhaps because it is the obvious choice and generally performs well, most studies are limited to self-selection. There is also no need to do otherwise, as these studies often start with fully labelled data and only simulate the labelling; this makes it easy to redo the selection with a different classifier.

The other scenario, *foreign-selection*, where the selector and consumer are different, has received less attention than it needs. The scenario may seem counter-intuitive at first, but there are many practical situations where the samples selected for one classifier are used to train another classifier. For example, creating a labelled training set may be a large investment that is only worth the expense if the data can be reused for many different applications – maybe even for future applications that are not yet known when the selection is made. Another reason to use foreign-selection is efficiency: the final model may be so complex that it is too expensive to retrain it for each selection step. In that case a simpler, faster model may be used for the selection phase (e.g., Lewis and Catlett, 1994). And, as a final example, in some applications it is only possible to select the best way to model for the problem *after* the examples have been selected – for example, in natural language processing it is not uncommon to collect and label the data first and only then decide on the best language model (Baldridge and Osborne, 2004).

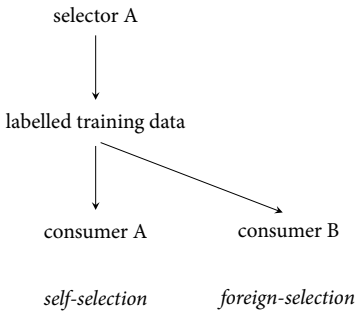


Figure 2: Active learning is a two-step process: first the samples are selected, then a final classifier is trained. In self-selection the same classifier is used for the selector and the consumer. In foreign-selection the consumer is a different classifier.

Using the samples selected for one classifier to train another is a form of *sample reuse*. Sample reuse is always possible, but is it also a good idea? In any active learner, the sample selection is based on the preferences of the classifier used in the selection; it is far from obvious that a completely different classifier will be able to learn from the same sample selection. *Sample reusability* indicates the scenario where sample reuse is successful. There is said to be sample reusability if the consumer learns more, or at least not less, from the active selection than it would learn from a random selection of the same size.

Sample reusability is important for the decision to use active learning. Active learning is useful if it improves the model performance and reduces the number of labelled samples. But if foreign-selection is necessary and there is no sample reusability, it would be better not to use active learning.

The main reason why sample reusability can be a problem is the bias in the active sample selection. Active learning makes a sample selection that is biased towards the selector: compared with a random selection of samples, the active selection includes more samples in areas that the

selector finds interesting and fewer in other areas. If the consumer does not share the selector's preferences, there may be areas that the selector thought uninteresting but that the consumer finds very important. In that case the consumer will find better information about these areas in the random selection. The active selection might compensate with better information in other areas, but this will not always be enough. If the negative effects of the bias are larger than the positive effects, it may have been better not to use active learning.

Importance-weighted active learning (IWAL) is a recent active learning strategy that aims to reduce the bias in the sample selection. It solves the problems of uncertainty sampling and other simple active learning strategies by combining a biased random selection process with importance weighting. Because of the importance weights, importance-weighted active learning provides unbiased estimators for the prediction error. This is a useful property that might improve the sample reusability. In fact, the authors of the importance-weighted active learning framework claim that because the estimates are unbiased, the algorithm *will* produce a reusable sample selection (see Beygelzimer et al., 2011, and the related presentation). This paper investigates this claim and concludes that it is not true: unbiased estimators help, but they are not enough to guarantee sample reusability.

The rest of this paper explores these topics in more detail. Section 2 gives a definition of sample reusability and provides a short overview of earlier research. Section 3 introduces importance-weighted active learning; section 4 explains how it corrects the bias in the sample selection and why this solves part of the reusability problem. Section 5 shows that bias correction is not enough and discusses the important differences between random and active sample selections. Section 6 argues that there is no universal reusability. Section 7 describes experiments with the reusability of importance-weighted active learning in practice. There may still be reusability between some classifier pairs, so section 8 tries to find the necessary conditions.

2 Sample reusability

An active learner makes a sample selection for a particular classifier, the selector, but the final result does not have to be an instance of that same classifier. Perhaps the samples will be used to train a different type of classifier, the consumer, or perhaps the samples are even used for something other than classification. These are examples of *sample reuse*: reusing the sample selection for something for which it was not selected. Tomanek (2010) gives a more formal definition:

Definition 2.1 (Sample reuse)**.** Sample reuse describes a scenario where a sample S obtained by active learning using learner T_1 is exploited to induce a particular model type with learner T_2 with $T_2 \neq T_1$.

Sample reuse is always possible; the question is how well it works. The sample selection is based on the preferences of the classifier that makes the selection, so it is acceptable for foreign-selection to perform somewhat

worse than self-selection. But at least foreign-selection should give a better performance than learning from a random sample: if sample reuse is worse than random sampling, it would be better not to use active learning at all. Therefore, we speak of *sample reusability* if we expect that the consumer will learn more from the samples selected by the selector than from a random sample selection. This is reflected in the definition of sample reusability by Tomanek (2010):

Definition 2.2 (Sample reusability)**.** Given a random sample S_{RD}, and a sample S_{T_1} obtained with active learning and a selector based on learner T_1, and a learner T_2 with $T_2 \neq T_1$. We say that S_{T_1} is reusable by learner T_2 if a model θ' learned by T_2 from this sample, i.e., $T_2\,(S_{T_1})$, exhibits a better performance on a held-out test set $\mathcal{T}$ than a model θ'' induced by $T_2\,(S_{RD})$, i.e., $\text{perf}\left(\theta', \mathcal{T}\right) > \text{perf}\left(\theta'', \mathcal{T}\right)$.

Note that this definition of sample reusability is subject to chance. It depends on the initialisation and samples presented to the algorithm. In this paper, sample reusability means *expected* sample reusability: does the algorithm, averaged over many runs, perform better with active learning than with random sampling? That is, there is sample reusability if

$$\text{E}\left[\text{perf}\left(\theta', \mathcal{T}\right)\right] > \text{E}\left[\text{perf}\left(\theta'', \mathcal{T}\right)\right]$$

Early active learning papers occasionally mentioned sample reusability – for example, Lewis and Catlett (1994) discuss "heterogeneous uncertainty sampling" – but overall the problem has received little attention. There is still no clear answer to why, when and whether foreign-selection works. In the most extensive study so far, Tomanek and Morik (2011) formulated and tested a number of hypotheses about foreign-selection with different classifiers and datasets. The results are inconclusive: foreign-selection sometimes works and sometimes does not, and there are no classifier combinations that always perform well. None of the hypotheses, such as "similar classifiers work well together", could be confirmed. Experiments by Baldridge and Osborne (2004) and by Hu (2011) show similar results.

These studies have an important limitation: they only discuss uncertainty sampling, one of the simplest active learning strategies. Uncertainty sampling has well-known problems (Dasgupta and Hsu, 2008) that could explain some of the results of the reusability experiments: it creates a strongly biased sample selection, which is a problem since some classifiers are very bias-sensitive (Zadrozny, 2004; Fan and Davidson, 2007), and does nothing to reduce this bias. With foreign-selection the bias may be even worse, because it was created for a different type of classifier.

The reusability results are probably not independent of the selection strategy. If the bias is one reason why uncertainty sampling does not produce reusable results, a selection strategy with a weaker bias might improve the sample reusability. The next section will discuss importance-weighted active learning, a recent selection strategy that solves several bias-related problems. Because it uses importance weighting to correct the bias in the sample selection, it might also produce more reusable sample selections.

3 *Importance-weighted active learning*

Importance-weighted active learning (Beygelzimer et al., 2009) appears to solve many of the problems of earlier active learning strategies, because it combines importance weighting with a biased random selection process. Importance weighting helps to correct the bias, while the randomness in the selection ensures that the does not systematically exclude any area of the sample space. Together, these two properties make it possible to prove that importance-weighted active learning will converge to the same solution as random sampling (Beygelzimer et al., 2010).

The importance-weighted active learning algorithm (Figure 3) is a sequential active learner: it considers each example in turn, and decides immediately if this new example should be labelled or not. The algorithm uses a "biased coin toss" to decide if it should label the example, with a selection probability P_x that defines the probability that example x will be labelled. If an example is selected, it is labelled and added to the labelled dataset with a weight set to the inverse of the selection probability (the next section will explain why this is a useful choice).

For each new example x:

1. Calculate a selection probability P_x for x.
2. With probability P_x: query the label for x and add x to the labelled dataset with importance weight $w_x = \frac{1}{P_x}$.

Figure 3: The importance-weighted active learning algorithm (Beygelzimer et al., 2009).

Different implementations of importance-weighted active learning have different definitions for the selection probability, but they share the same idea: the probability should be higher if the new example is likely to be interesting, and lower if it is not. The number of samples in the labelled dataset depends on the definition of the selection probability. An implementation with higher selection probabilities will select more samples than an implementation that tends to choose smaller selection probabilities.

It is important that the selection probabilities are always greater than zero. This guarantees that the algorithm will eventually converge to the optimal hypothesis and does not show the missed cluster problem. The missed cluster problem (Dasgupta and Hsu, 2008) explains how simple active learners can end up in a local optimum and can produce a classifier that is very far from the optimal classifier.

For example, consider uncertainty sampling, the simple strategy that selects the example that is closest to its current decision boundary. This selection strategy is likely to produce missed cluster problems (Figure 4). Because it only collects examples near its initial guess, it will never explore the rest of the sample space: it might miss important samples. If the initial guess is close to a local optimum, the algorithm will select examples that bring it closer to that local optimum. In importance-weighted active learning, the non-zero selection probabilities guarantee that the algorithm explores the complete sample space and will eventually converge to the optimal hypothesis.

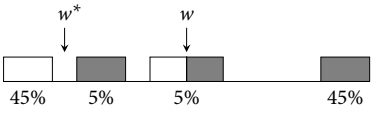


Figure 4: The missed cluster problem: if the initial samples are drawn from the two groups in the middle, close-to-the-boundary sampling will keep querying samples near the initial boundary w. Because it never looks elsewhere, the classifier will never find the optimal boundary w^*. (Dasgupta and Hsu, 2008)

Beygelzimer et al. (2010) use the following definition of the selection probability P_k of the kth sample:

$$P_k = \min\left\{1, \left(\frac{1}{G_k^2} + \frac{1}{G_k}\right) \cdot \frac{C_0 \log k}{k-1}\right\} \quad \text{where} \quad G_k = \text{err}\left(h'_k, S_k\right) - \text{err}\left(h_k, S_k\right)$$

It compares the error of two hypotheses h_k and h'_k, on the current set of labelled samples S_k. h_k is the current hypothesis, that is: the hypothesis that minimises the error on S_k. h'_k is the alternative hypothesis: the hypothesis that disagrees with h_k on the label of the sample k, but otherwise still minimises the error on S_k. C_0 is a free parameter that scales the probabilities: choosing a larger C_0 will select more samples.

The errors of the two hypotheses give an idea of the quality of the current prediction. The current hypothesis is optimal on the current set of labelled samples, but it is not necessarily optimal on the true distribution. The alternative hypothesis cannot be better than the current hypothesis on the current set, but it can still be better on the true distribution.

There are two possibilities. Suppose that on the true data, the alternative hypothesis is better than the current hypothesis. In that case the errors of the two hypotheses on the current set are likely to be close together, since the current hypothesis can never be too far off – it converges to the optimal solution, after all. Suppose, on the other hand, that the current hypothesis is indeed better than the alternative, even on the true data. In that case the error of the alternative hypothesis on the current data is likely to be higher than that of the current hypothesis.

The difference between the two errors predicts how valuable the new example could be. If the difference is small, it is likely that the current hypothesis is wrong. The current set of data does not provide enough information on this new sample, so it is useful to ask for the label. If the difference is large, it is likely that the current hypothesis is correct. The current set of data has evidence to support the current label of the new example, so it is probably not useful to ask for the label.

Beygelzimer et al. (2010) use this intuition in their definition of the selection probability: a larger difference in error leads to a smaller selection probability. Note also that the selection probabilities become smaller as the number of samples grows: the more samples seen, the more evidence there already is to support the current hypothesis. In their paper, Beygelzimer et al. provide confidence and label complexity bounds for their definition. As expected, there is a trade-off between the quality of the prediction and the size of the sample selection. Large C_0 provide large sample selections and results similar to random sampling, whereas small C_0 provide smaller sample selections that are sometimes much worse than random selections.

4 Bias correction with importance weights

Like any active learning algorithm, importance-weighted active learning creates a biased sample selection: there are more samples from areas that are interesting to the selector, and fewer from other areas. This bias cannot be prevented – active learning is only effective if it can skip certain examples – but there are ways to correct the bias before training a classifier.

Importance weighting is one way to correct the bias in a sample selection. Importance-weighted active learning assigns a weight w_x to each example in the selection: larger weights for examples from undersampled areas and smaller weights for the examples that have been oversampled. Importance-weighted versions of classifiers use these weights in their optimisations. For example, the importance-weighted zero-one classification error could be defined as the sum of the normalised weights of the misclassified samples. If the importance weights are set to the correct values, this importance-weighted estimator is an unbiased estimator for the classification error.

The quality of importance weighting depends on the correct choice of the importance weights. One option is to derive the weights from density estimates, by comparing the density of the sample selection with that of the true distribution. Making this estimate is difficult, so this method can lead to variable and unreliable results.

Importance-weighted active learning has an advantage: because it makes a biased-random sample selection with a known selection probability, the algorithm knows exactly how biased the selection is. By using the inverse of the selection probability as the importance weight, importance-weighted active learning can calculate the perfect correction to its self-created bias. Beygelzimer et al. (2010) prove that the importance-weighted estimators of importance-weighted active learning are indeed unbiased.

To see how this works, compare the probability density of an example in random sampling with that same probability density in importance-weighted active learning. In random sampling, the probability density of x is equal to its density in the true distribution: $P_{RD}(x) = P(x)$. In importance-weighted active learning, the probability density of x in the labelled set depends on two things: the density in the true distribution, i.e., the probability that the sample is offered to the algorithm, and the probability $s(x)$ that the algorithm decides to label the example: $P_{IWAL}(x) = P(x) \cdot s(x)$. It gives the example the importance weight $\frac{1}{s(x)}$, so the expected importance-weighted density of x is the same in both algorithms: $P_{RD}(x) = \frac{1}{s(x)} \cdot P_{IWAL}(x) = P(x)$ and the sample selection is unbiased.

With its 'perfect' importance weights, importance-weighted active learning can perhaps provide a better sample reusability than unweighted active learning strategies. The skewed, biased density distribution of the active selection can be corrected into an unbiased estimate of the true density distribution. This removes one component that limits sample reusability.

With the importance weighted correction, the density of the samples selected by importance-weighted active learning converges to the density that would be expected from random sampling. Given an unlimited supply of samples, the two density distributions will eventually become very similar.

Unlike unweighted active learning, where the bias alone can be enough to prevent reusability, importance-weighted active learning creates an unbiased sample selection that might be more reusable. For this reason, it seems that importance-weighted active learning should always be preferred to active learning without importance weights.

5 *Are the sample distributions the same?*

While importance weighting corrects the bias and produces unbiased estimators, there are still two important differences between the active selection and the random selection. One, the unweighted distribution of the samples is different, so the active selection has different levels of detail than the random selection. Two, the importance weights introduce an extra source of variance in the density distribution.

Although the correction makes the averaged, importance-weighted density of the active sample selection equal to the density of the random sample selection and the true distribution, the unweighted density is different. In importance-weighted active learning, the unweighted probability density of x, $P_{IWAL}(x)$, depends on the selection probability $s(x)$ that is determined by the algorithm. On average, compared with random sampling and the true distribution, the active sample selection will have relatively more examples for which $s(x)$ is large and relatively fewer examples for which $s(x)$ is small. This is normal, since the active learner would not be an active learner if it did not influence the sample selection.

A simple experiment shows that an importance-weighted active learner has a preference for examples that are interesting to its classifier. The Vowpal Wabbit is an implementation of the importance-weighted active learning algorithm, with a selection probability similar to that of Beygelzimer et al. (2010). Figure 5 shows the sample selection of the Vowpal Wabbit on a simple two-class problem with uniformly distributed samples. Random selection follows the uniform density, but the active sample selection is different. The unweighted density of samples near the decision boundary at $x = 0$ is much higher in the active selection than it is in the random selection (top). This cannot be repaired. Importance weighting gives the correct weighted probability density (bottom), but the absolute number of selected examples is still different. There are more examples in some areas and fewer examples in others, relative to a random selection of the same size.

The sampling priorities affect the level of detail: there will be more detail in areas with more examples, but there will also be less detail in areas with fewer samples. This could be a problem if the underrepresented areas are of interest to the classifier: if the active selection has fewer details there, the classifier might be better off with a random sample.

Because of this difference, sample reusability cannot be guaranteed. The lack of detail means that for every problem, there is probably a pair of classifiers that does not work together. This leads to the conclusion that there is no universal reusability. This is discussed in section 6. In some cases, there may still be reusability if two classifiers are interested in the same details. Section 8 discusses some conditions that guarantee this.

There is another source of trouble: the importance weights increase the variance in the sample distribution. The importance-weighted distribution may be similar to the random distribution *on average*, but in a single run of the algorithm the sample selection can be very dissimilar. The algorithm gives large importance weights to samples with a small selection probability,

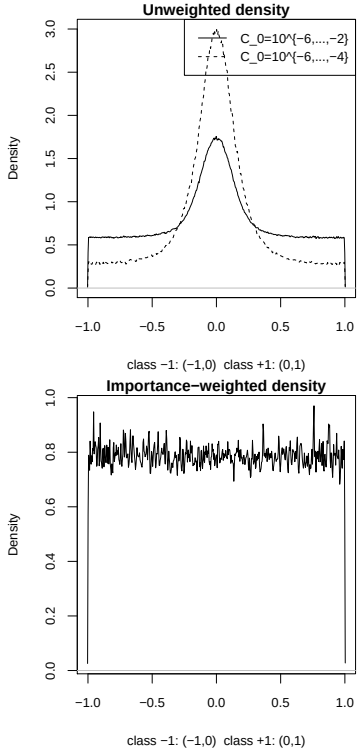


Figure 5: Two plots of the density of active learning selections made by the Vowpal Wabbit (1000 runs with 1000 unlabelled examples, for various C_0). The 1D problem has two uniform classes, class -1 at $x = [-1, 0)$ and class $+1$ at $x = [0, 1]$. The importance-weighted density (bottom) follows the true class distribution, but the unweighted density (top) shows that the algorithm selects most examples from the area around the decision boundary. The peak in the middle is more pronounced if the algorithm can select fewer examples (smaller C_0); most queries outside that area were made with the less aggressive settings (larger C_0).

to compensate for their relative undersampling in the average dataset. However, in the few datasets where one or more of these rare samples are selected, their large importance weight will give them a disproportionally strong influence. This effect averages out when the number of samples increases, but it could be a problem with small sample sizes.

Perhaps the problem can be illustrated with a practical experiment. Such an experiment would need a dataset with outliers that 1. cause a significant change in the classifier, 2. are so rare and outlying that they are not a problem for random sampling, but 3. are still common enough to be picked by the active learning algorithm often enough to be a problem. Requirements 2 and 3 are contradictory: the outliers must be rare and frequent at the same time. One way to achieve these goals is to spread a large number of outliers over a large area. Individually they are outliers, but there are enough to ensure that one or two will be selected.

Consider this experiment on a dataset with circular outliers (Figure 6), with the linear Vowpal Wabbit as the selector and a consumer based on quadratic discriminant analysis (QDA). The dataset has two dense clusters in the middle and a circle of outliers orbiting those two clusters. The outliers have a label that is the opposite of the label of the closest cluster. After tuning the number of samples in the circle, it is possible to find a distribution where the QDA classifier trained on the active samples performs consistently worse than the QDA classifier trained on random samples (Table 1). Closer inspection of individual runs shows that many cases, the QDA classifier is thrown off-balance by an outlier with a heavy weight.

It is quite hard to trigger this behaviour on purpose: it does not happen at small sample sizes, the density of the outliers has to be just right, and this distribution does not cause problems for linear discriminant analysis or linear support vector machines. Still, this example illustrates that the importance weights can sometimes introduce new problems.

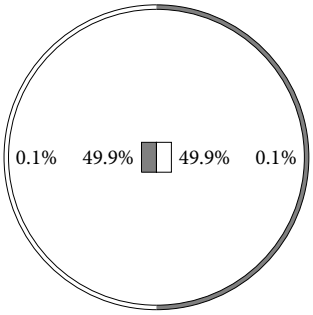


Figure 6: The 'circle' dataset: a 2D problem with two dense clusters and a very sparse circle with samples from the opposite class.

# unlab.	IWAL error	Random error
10	0.1573237	0.1565162
50	0.05967161	0.06077859
100	0.05496723	0.05203855
500	0.04241635	0.02837003
1000	0.0372564	0.02339909
2500	0.03063014	0.02009514
5000	0.02538832	0.01889917
10000	0.02075104	0.01631644

Table 1: Errors of QDA with IWAL and random sampling, on the circle dataset with circle density 0.001, for different numbers of available (unlabeled) examples. The mean error with IWAL was significantly higher than with random sampling.

6 *Consequences for reusability*

The lack of detail in some areas of an importance-weighted active selection has consequences for the reusability of that selection. In the areas that are undersampled by active learning, the random selection can provide more detail than the active selection. This is important if the consumer depends on the information from those areas. If the active selection provides less detail than the random selection, the consumer might have learned more from the random selection: sample reusability cannot be guaranteed.

Whether this problem occurs in practice depends first of all on the correspondence between the selector and the consumer. If both classifiers are interested in the same areas of the sample space, the examples will also be useful to both classifiers. Even if only some parts of the areas of interest overlap, active learning could still be better if the improvement in one area is large enough to compensate for the loss in another area.

The effect also varies with the sample size. At small sample sizes the lack of detail can matter. At larger sample sizes the difference becomes less noticeable – there are more samples even in undersampled areas – but the active selection may still have less detail than a random sample of equal size.

Since there are almost always some consumers that need details that the selector did not provide, the conclusion must be that sample reusability cannot be guaranteed. The lack of detail in undersampled areas means that there is always the possibility that a consumer would have learned more from a random than from an active sample.

It is possible to construct an experiment that shows this behaviour. Choose a selector and a consumer, such that the consumer can produce all of the solutions of the consumer, plus some solutions that the selector cannot represent. More formal: the hypothesis space of the selector should be a proper subset of the hypothesis space of the consumer. Next, construct a classification problem where the optimal solution for the selector is different from the optimal solution of the consumer. Run the experiment: use importance-weighted active learning to select samples, train a consumer on the active selection and a consumer on a random selection of the same size. Measure the errors of both classifiers on a held-out test set. For small sample sizes, expect the performance of a consumer trained on a random sample selection to be better than the performance of the consumer trained on the active selection.

For example, consider a one-dimensional two-class dataset (Figure 7) with four clusters in a $+ - + -$ pattern: a tiny cluster of the $+$ class, a large cluster of $-$, an equally large cluster of $+$ and a tiny cluster of $-$. The optimal decision boundary for a linear classifier is in the middle between the two large clusters. It will misclassify the two tiny clusters, but that is inevitable. A more sophisticated classifier can improve on the performance of the simple classifier if it assigns those clusters to the correct class.

Figure 7: A simple 1D dataset. Most of the samples are in the middle clusters, each of the clusters on the side has 1% of the samples.

Active learning with the linear classifier as the selector selects most samples from the area near the middle, since that is where the linear classifier can be improved. As a result, the active sample selection does not have many examples from the tiny clusters. This makes it harder for the sophisticated classifier to learn the correct classification for those areas. With the random selection, where the samples from the tiny clusters have not been undersampled, the sophisticated classifier would have a better chance to assign the correct labels to the tiny clusters. In this example the expected performance with random sampling is better than with active learning: the sample selection from active learning is not reusable.

Plots of the learning curve (Figure 8) show this effect in an experiment with the linear Vowpal Wabbit as the selector and a support vector machine with a radial basis function kernel as the consumer. The plots show the results of experiments with different values for the parameter C_0, used in the definition of the selection probability in the Vowpal Wabbit. As follows from the definition of the selection probability, the parameter C_0 determines the aggressiveness of the active learner. If C_0 is very small, the selection probabilities will be smaller: the number of selected samples will be small, too. For larger C_0, the selection increases and so does the number of selected samples. At some point the selection probabilities are so large that the active learner will include every sample in its selection, and there is no longer any difference between the active and random selections. These extreme cases are not representative for real active learning problems.

As predicted, the selection by the simple linear selector is of limited use to the more complex radial-basis support vector machine. Foreign-selection with this combination of classifiers leads to results that are worse than those of random sampling. The effect is at its strongest at small sample sizes, but is still noticeable at larger the sample sizes. The number of labelled samples can increase for two reasons: because the number of unlabelled examples is larger, or because the C_0 parameter is higher. In both cases the number of samples from undersampled areas increases and the consumer will receive more information to make a good decision on the samples in the corners. Beyond this crossover point the active learner could perform better than random sampling, even if it undersamples some important areas, because the extra efficiency in the other areas more than compensates for the loss of precision. Whether and when this crossover point occurs depends on the circumstances.

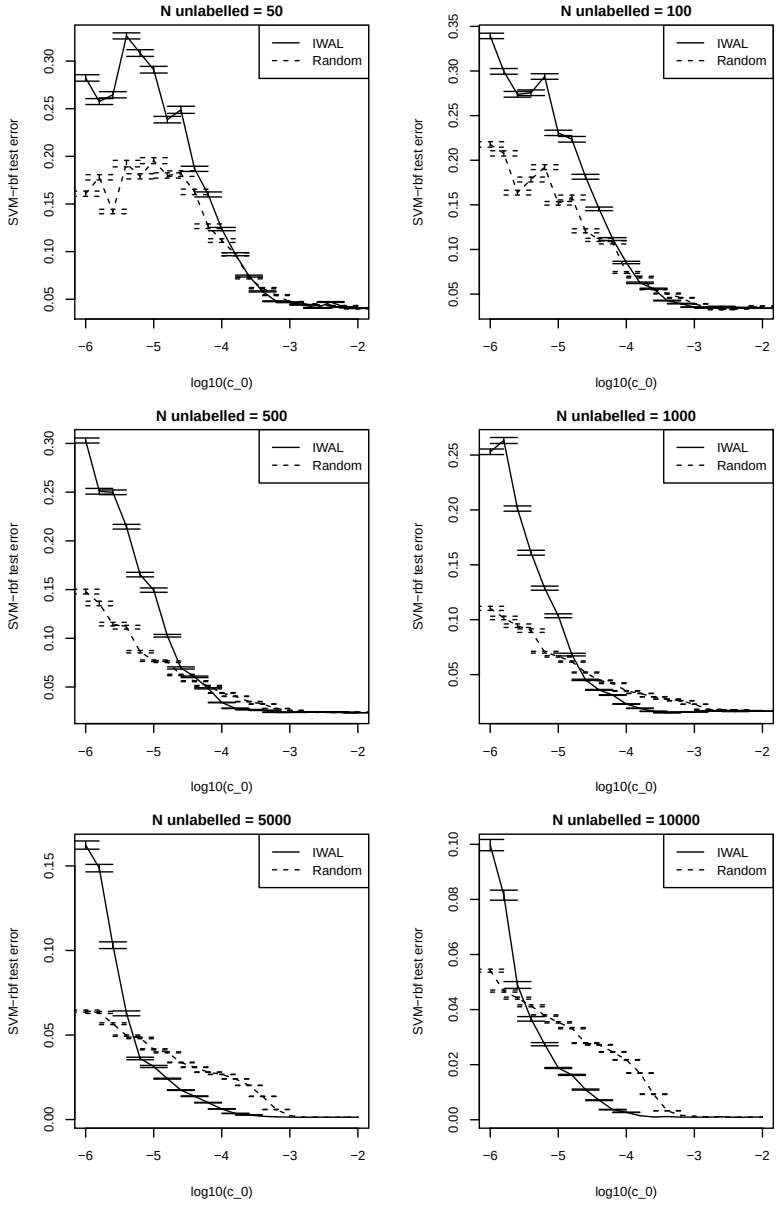


Figure 8: The mean test error of a radial basis support vector machine classifier trained on samples selected by the Vowpal Wabbit importance-weighted active learning (IWAL) algorithm. The error bars indicate the standard deviation of the means. The dataset is the dataset shown in Figure 7.

7 Experiments

Do the theoretical problems discussed in the previous sections also occur in practice? This section presents the results of reusability experiments I did with several classifiers on five datasets, from the UCI Machine Learning Repository and the Active Learning Challenge 2010. I used three selection strategies: random sampling, uncertainty sampling and importance-weighted active learning. To evaluate the contribution of importance weighting to the results, I also looked at importance-weighted active learning without importance weights, using the same sample selection but with the importance weights set to 1.

The datasets I used are not representative for real-world problems, and neither are these experiments. These experiments are not intended to make any general predictions about reusability in practical applications – making such predictions would be very hard, if not impossible. Rather, these experiments are intended to discover if the reusability problems discussed before and demonstrated with hand-crafted datasets, also occur with independent datasets. The UCI datasets may be unrepresentative and sometimes synthetic, but at least they are not *designed* to cause reusability problems.

Note that the results I present here are the results for foreign-selection. The active learning results on the following pages may seem disappointing, but they do not show the results for self-selection; the results of importance-weighted active learning with self-selection may or may not be different.

In the rest of this section, I first provide more detail about the datasets, the sample selection strategies, the classifiers and the procedure I followed for my experiments. Then I discuss the results, illustrated by the most interesting of the learning curve graphs.

Datasets

I used five datasets for these experiments. Three datasets come from the UCI Machine Learning Repository (Frank and Asuncion, 2010): `car`, `bank` and `mushroom`. Two other datasets come from the Active Learning Challenge 2010 [1]: `alex`, a synthetic dataset, and `ibn_sina`, from a real-world handwriting recognition problem. Table 2 shows some statistics about these datasets. All datasets are two-class classification problems.

[1] `http://www.causality.inf.ethz.ch/activelearning.php`

Dataset	Source	Features	Examples	Positive	Test proportion
car	UCI	6, categorical	1728	30.0%	10%
bank	UCI	16, categorical, numerical	4521	11.5%	10%
mushroom	UCI	20, categorical	8124	51.8%	20%
alex	Active Learning Challenge	11, binary	10000	73.0%	20%
ibn_sina	Active Learning Challenge	92, binary, numerical	20722	37.8%	20%

Table 2: Statistics of the datasets used in these experiments.

The `car` evaluation dataset is a somewhat synthetic dataset. The examples were derived from a hierarchical decision model, so there is exactly one example for each feature combination. This makes it an unrealistic choice for active learning, since active learning tries to exclude examples that are similar and depends on the density distribution of the samples.

In this dataset there are no duplicates and the density distribution may be distorted. The `alex` dataset is also synthetic, but in a different way: it is a toy dataset generated with a Bayesian network model for lung cancer. This might mean that the density distribution is closer to that of real data.

The datasets are small, perhaps too small for active learning. Active learning expects large datasets and the analysis often assumes an unlimited supply of unlabelled data. The Ibn Sina (`ibn_sina`) dataset is perhaps the closest to a real active learning problem: it is a very large dataset that contains real data, based on a handwriting recognition problem where Arabic words must be recognised in ancient manuscripts.

Sample selection

In these experiments, I compared random sampling, uncertainty sampling and importance-weighted active learning. The only practical implementation of importance-weighted active learning is the Vowpal Wabbit[2], a fast open-source program for online learning. The Vowpal Wabbit creates linear classifiers with or without active learning. I used a modified version for my experiments: one that produces sample selections instead of classifiers and that includes uncertainty sampling as an alternative active learning strategy.

The selector in these experiments is always a linear classifier, the only type the Vowpal Wabbit has. The selection strategy is either random sampling, uncertainty sampling or importance-weighted active learning. For uncertainty sampling it is easy to get a sample selection of a certain size: simply pick the first n samples from the selection. In importance-weighted active learning the number of samples depends on the parameter C_0 and on chance, but the random component of the selection means that the actual number of samples varies between different iterations (Figure 9).

From the three selection strategies, only importance-weighted active learning uses importance weighting. To determine the effect of the importance weights, I copied the selections from importance-weighted active learning and set the importance weights to 1 for all examples. This selection strategy is listed as 'IWAL (no weights)' in the graphs.

[2] `http://hunch.net/~vw/`

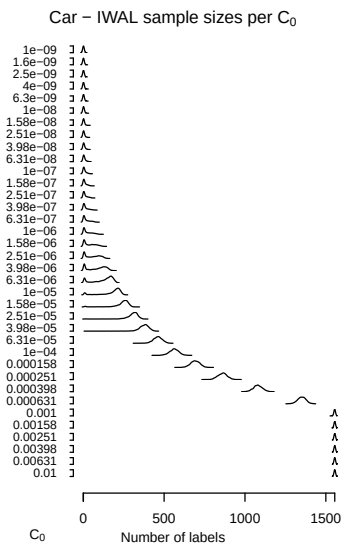


Figure 9: The sample selection size of importance-weighted active learning depends on the parameter C_0, but it is also a random variable. This graph shows how the average sample size for the `car` dataset increases for higher values of C_0. I chose the range of C_0 large enough to include both the very small and the very large sample selections. In practice you would choose a C_0 from the middle.

Classifiers

The selector is always a linear classifier, but I used a larger range of classifiers for the consumer. I used R (R Development Core Team, 2012) to experiment with six classifiers, all with support for importance weights. Most classifiers are from the `locClass` package (Schiffner and Hillebrand, 2010). I use their R function name to identify them:

- Linear regression (`lm`), probably the most similar to the linear model in the Vowpal Wabbit. This would be the `fisherc` in PRTools.
- Linear discriminant analysis (`lda`), also a linear classifier but based on different principles.
- Quadratic discriminant analysis (`qda`).
- Support vector machines (`wsvm-linear`), using a third approach to linear classification.

- Support vector machines with a third-degree polynomial kernel (`wsvm-poly3`).
- Support vector machines with a radial-basis kernel (`wsvm-radial`).

Unfortunately, not every classifier worked on every dataset. Especially the LDA and QDA classifiers often complained about singular data. In these cases there are no results for these classifier/dataset pairs.

In some cases the sample selection only had samples from one class, something that is more likely to happen with smaller sample sizes. Since that made it impossible to train a classifier, I removed these iterations from the results.

Implementation

I did the experiments with the Vowpal Wabbit for the selection and the R package for the final classifiers. The experiments followed this procedure:

1. For each iteration, before each selection, split the data in a training and test set.
2. For the random selection (Random): shuffle the training set and select the first n samples, for n at fixed intervals from only a few samples to every available sample.
   For the uncertainty selection (US): use the Vowpal Wabbit to rank the samples. Select the first n samples, for n from small to large.
   For importance-weighted active learning (IWAL): use the Vowpal Wabbit with C_0 values from 10^{-9} to 10^{-2}. For the datasets in these experiments this range of C_0 produces sample selections with only a few samples, selections that include every available sample and everything in between.
   For IWAL without weights, use the IWAL selection but with weights set to a constant 1.
3. Train each of the classifiers on each sample selection. For each combination, calculate the error on the test set.

Results

On the following pages, I show a selection of the learning curves for these experiments. The lines show the mean classification error on the test set, the semi-transparent bands indicate the standard deviation of the mean.

For random sampling and uncertainty sampling these plots are straightforward: I calculate the mean for each value of n. The calculations for importance-weighted active learning are more complicated. In importance-weighted active learning the number of samples is a random variable: the sample sizes are spread out (Figure 9). There is seldom more than one value at a specific sample size. To get useful data points for these experiments, I grouped the results for the same C_0 and calculated the mean and standard deviation. I show the group means at the median sample size for that C_0.

When interpreting these graphs, be aware that the results at the maximum sample sizes may not representative for the behaviour of the algorithms on real data. In these experiments there is a limited amount of data, so it is possible to select and label every example. In practice this would be

different, as there would be more samples than could ever be labelled. On the following pages, most learning curves end in the same point. When every sample is selected there is no difference between learning strategies. For this reason it may be best to look at the middle section of each graph: more than one or two samples, but less than the maximum number of samples.

The learning curves of the polynomial and radial-basis support vector machines on the Ibn Sina dataset (Figure 10) show the result that you hope for. The sample selection of importance-weighted active learning are very reusable for the polynomial kernel: the samples are more useful than those of random sampling. The radial-basis kernel does not perform better with active learning, but also not much worse; uncertainty sampling, however, does not work well. These are good reusability results for importance-weighted active learning. It is hard to say why these classifiers show these results: apparently, on this dataset, they are interested in similar examples.

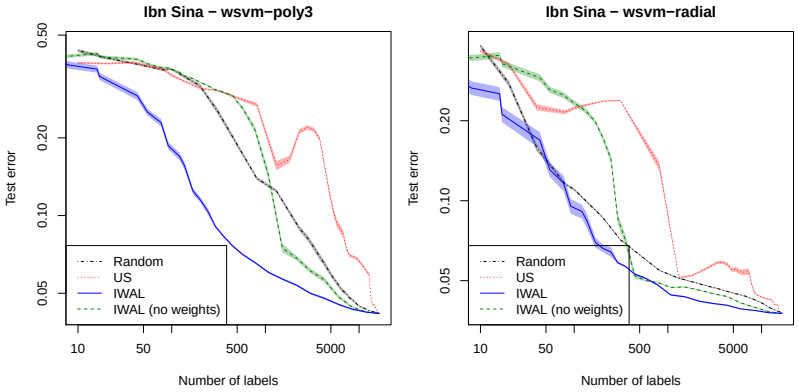


Figure 10: Positive results of importance-weighted active learning on the Ibn Sina dataset with polynomial and radial-basis support vector machines.

In other cases the result of importance-weighted active learning is close to that of random sampling (Figure 11), while uncertainty sampling is worse. The results of uncertainty sampling are even worse than in Ibn Sina results shown above. This could be due to the missed cluster problem: the shape of the line suggests that uncertainty sampling is exploiting a local optimum, and only selects examples from other areas after it has labelled every example around its initial decision boundary. Importance-weighted active learning, on the other hand, looks much safer: it is not better than random sampling, but also not much worse.

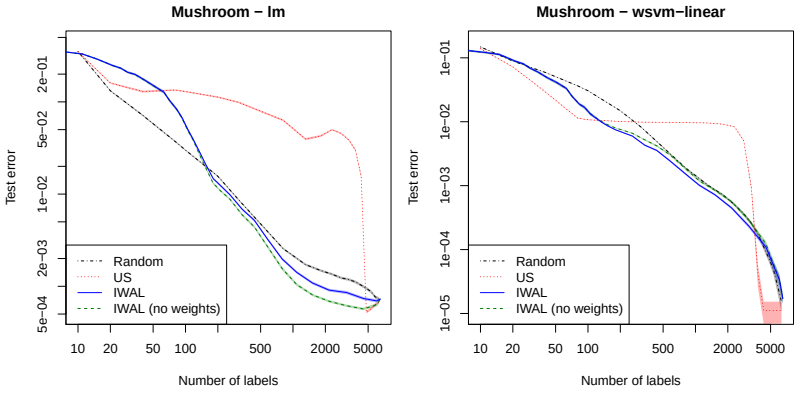


Figure 11: On the mushroom dataset, importance-weighted active learning has an advantage over uncertainty sampling, which may have found an instance of the missing cluster problem. Similar behaviour can be seen in other datasets.

However, importance-weighted active learning can also perform worse than the other sampling methods (Figure 12). This shows that the sample selection by importance-weighted active learning is not always reusable: the result that was predicted earlier in this paper.

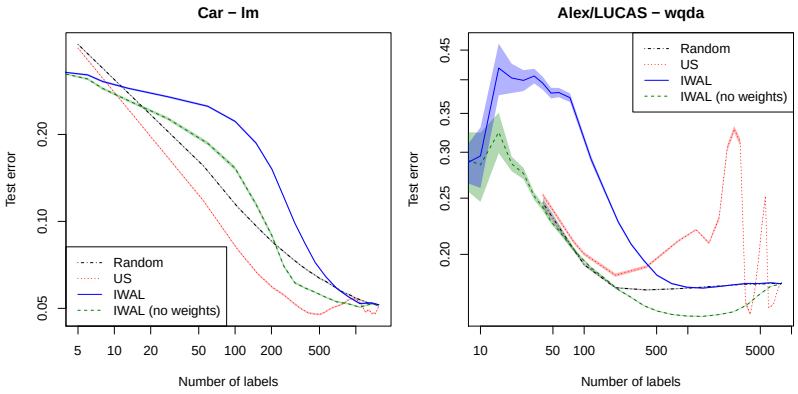


Figure 12: Sometimes importance weighting is the problem: removing the weights from the importance-weighted selection improves the results. This happens most often with LDA/QDA and on the Alex dataset and could be due to the instability that is introduced by the large importance weights.

The graphs in figure 12 show another interesting result: importance weighting is not that good. There are quite a few examples where it helps to remove the importance weights. This often happens with LDA and on the Alex dataset. Perhaps this is a result of the variability from the large importance weights, the problem discussed earlier in this paper. It is curious to see that when there is no reusability for importance-weighted active learning, the unweighted selection often *does* show reusability. Unfortunately, removing the importance weights is not an option: there are also examples where importance-weighted active learning does work but the unweighted version does not (e.g., Figure 10).

From these results, it becomes clear that importance-weighted active learning is not a solution for the reusability problem. There are certainly cases where the samples are reusable. From the experiments discussed here, one could even get the impression that it is reusable in more cases than uncertainty sampling. However, there are too many examples where the selection from importance-weighted active learning is not reusable to maintain that it solves the reusability problem.

A second, somewhat discouraging conclusion from these experiments is that importance weighting is not always helpful. The bias correction may be correct on average, but in individual sample selections it is imprecise. As predicted in the previous sections, the variability that is introduced by the large importance weights sometimes leads to a performance that is much worse than the performance without the weights.

8 *Conditions for reusability*

The theoretical discussion and the experiments show that there is no sample reusability between *every* pair of classifiers on *every* possible problem, but also that there are combinations where there is sample reusability. When can a selector-consumer pair guarantee good results? This section explores possible conditions for sample reusability.

Hypothesis spaces and hypotheses

Let $\mathcal{H}_{sel}$ and $\mathcal{H}_{cons}$ be the hypothesis spaces of the selector and consumer, respectively. The hypothesis space is the set of all hypotheses that a classifier algorithm can form: for example, the hypothesis space of a linear classifier on a two-dimensional problem is the set of all lines. Let $\text{err}(h, S)$ denote the (negative) performance of a hypothesis h on a sample selection S, where a smaller $\text{err}(h, S)$ is better. If S is an importance-weighted sample selection, $\text{err}(h, S)$ is the importance-weighted error. Let $\text{err}(h)$ be the expected performance of hypothesis h on new, unseen data from the true distribution.

Define the optimal hypothesis for the selector, h^*_{sel}, and the optimal hypothesis for the consumer, h^*_{cons}, as the hypotheses that minimise the expected error on unseen data:

$$h^*_{sel} = \arg\min \{\text{err}(h) : h \in \mathcal{H}_{sel}\}$$
$$h^*_{cons} = \arg\min \{\text{err}(h) : h \in \mathcal{H}_{cons}\}$$

Let $S_{AL,n}$ and $S_{RD,n}$ be the sample selections of n labelled samples, made by active learning (AL) and random sampling (RD), respectively. Note that for importance-weighted active learning, $S_{AL,n}$ includes importance weights and $\text{err}(h, S_{AL,n})$ is the importance-weighted error. Use $h_{sel,AL,n}$, $h_{sel,RD,n}$, $h_{cons,AL,n}$ and $h_{cons,RD,n}$ to denote the optimal hypotheses of the selector and consumer on these sample selections:

$$h_{sel,AL,n} = \arg\min \{\text{err}(h, S_{AL,n}) : h \in \mathcal{H}_{sel}\}$$
$$h_{sel,RD,n} = \arg\min \{\text{err}(h, S_{RD,n}) : h \in \mathcal{H}_{sel}\}$$
$$h_{cons,AL,n} = \arg\min \{\text{err}(h, S_{AL,n}) : h \in \mathcal{H}_{cons}\}$$
$$h_{cons,RD,n} = \arg\min \{\text{err}(h, S_{RD,n}) : h \in \mathcal{H}_{cons}\}$$

Assume that the classifier indeed minimises $\text{err}(h, S)$, that is, that it minimises the empirical risk and does indeed select these hypotheses when given $S_{AL,n}$ or $S_{RD,n}$.

Expected error and sample reusability

Before defining any conditions for reusability, we should take a closer look at the hypotheses that are selected by active learning and by random sampling. More specific, we should derive the expected error of these hypotheses, both on the active and random sample selections and on unseen data. Sample reusability is defined in terms of these expected errors.

First, note that random sampling gives unbiased estimates of the error: $\text{E}[\text{err}(h, S_{RD,n})] = \text{err}(h)$, the expected error of a hypothesis h on a random sample selection is equal to the expected error on unseen data. Or: the empirical risk averaged over all possible random sample selections is equal to the true error. This means that the optimal hypothesis h^*_{cons} – the hypothesis with the smallest expected error on unseen data – will also have the smallest expected error on the random sample selection:

$$\begin{aligned}\text{E}\left[\text{err}\left(h^*_{cons}, S_{RD,n}\right)\right] &= \text{err}\left(h^*_{cons}\right) \\ &= \min \{\text{err}(h) : h \in \mathcal{H}_{cons}\} \\ &= \min \{\text{E}[\text{err}(h, S_{RD,n})] : h \in \mathcal{H}_{cons}\}\end{aligned}$$

Importance-weighted active learning has the same property. Because it uses importance weighting, importance-weighted active learning produces an unbiased estimator of the error, i.e., $\mathrm{E}\left[\mathrm{err}\left(h, S_{AL,n}\right)\right] = \mathrm{err}\left(h\right)$ (Beygelzimer et al., 2009). The average $\mathrm{err}\left(h, S_{AL,n}\right)$ over all possible importance-weighted sample selections $S_{AL,n}$ is equal to the expected error on unseen data. (Note that $S_{AL,n}$ includes importance weights, so $\mathrm{err}\left(h, S_{AL,n}\right)$ is the importance-weighted error.) Then the expected error of the optimal hypothesis h^*_{cons} will also be optimal on the importance-weighted sample:

$$\begin{aligned}\mathrm{E}\left[\mathrm{err}\left(h^*_{cons}, S_{AL,n}\right)\right] &= \mathrm{err}\left(h^*_{cons}\right)\\ &= \min\left\{\mathrm{err}\left(h\right) : h \in \mathcal{H}_{cons}\right\}\\ &= \min\left\{\mathrm{E}\left[\mathrm{err}\left(h, S_{AL,n}\right)\right] : h \in \mathcal{H}_{cons}\right\}\end{aligned}$$

In both cases, with importance-weighted active learning and random sampling, the optimal hypothesis has the lowest expected error on the sample selection. This does not mean that the optimal hypothesis will also be selected. The hypotheses $h_{cons,RD,n}$ and $h_{cons,AL,n}$ are the hypotheses with the lowest error on the sample selection. The optimal hypothesis has the best *expected* error and it will have the smallest error if the sample size is unlimited, but in an individual sample selection with a limited size there may be another hypothesis that has a smaller error on the training set. This difference is relevant for sample reusability.

Since the selected hypotheses $h_{cons,RD,n}$ and $h_{cons,AL,n}$ are not optimal, they will have a larger expected error than the optimal hypothesis h^*_{cons}:

$$\begin{aligned}\mathrm{err}\left(h_{cons,RD,n}\right) &= \mathrm{err}\left(h^*_{cons}\right) + \varepsilon_{cons,RD,n}\\ \mathrm{err}\left(h_{cons,AL,n}\right) &= \mathrm{err}\left(h^*_{cons}\right) + \varepsilon_{cons,AL,n}\end{aligned}$$

where $\varepsilon_{cons,RD,n}$ and $\varepsilon_{cons,AL,n}$ represent the extra error introduced by the selection strategy and the sample size of random sampling and active learning.

Similarly, the hypotheses selected by the selector, $h_{sel,RD,n}$ and $h_{sel,AL,n}$ will also have an expected error that is higher than h^*_{sel}:

$$\begin{aligned}\mathrm{err}\left(h_{sel,RD,n}\right) &= \mathrm{err}\left(h^*_{sel}\right) + \varepsilon_{sel,RD,n}\\ \mathrm{err}\left(h_{sel,AL,n}\right) &= \mathrm{err}\left(h^*_{sel}\right) + \varepsilon_{sel,AL,n}\end{aligned}$$

where $\varepsilon_{sel,RD,n}$ and $\varepsilon_{sel,AL,n}$ represent the extra error introduced by the selection strategy and the sample size of random sampling and active learning.

If the active learner is functional, it should produce hypotheses that are better than random sampling, at least for the selector. For a functional active learner, the active hypothesis $h_{sel,AL,n}$ is expected to have a lower error than the random hypothesis $h_{sel,RD,n}$ for an equal sample size, so

$$\mathrm{err}\left(h_{sel,AL,n}\right) \leq \mathrm{err}\left(h_{sel,RD,n}\right)$$

which implies that $\varepsilon_{sel,AL,n} \leq \varepsilon_{sel,RD,n}$.

Following the definition of reusability, there is sample reusability with a consumer if the hypothesis produced by active learning is not worse than the hypothesis produced by random sampling:

$$\text{sample reusability if } \mathrm{err}\left(h_{cons,AL,n}\right) \leq \mathrm{err}\left(h_{cons,RD,n}\right)$$

There is sample reusability if $\varepsilon_{cons,AL,n} \leq \varepsilon_{cons,RD,n}$. Assume that the active learner is functional, i.e., that $\varepsilon_{sel,AL,n} \leq \varepsilon_{sel,RD,n}$. Does that also imply that $\varepsilon_{cons,AL,n} \leq \varepsilon_{cons,RD,n}$, i.e., that there is sample reusability?

The rest of this section tries to answer to this question, to formulate conditions that guarantee that there is sample reusability between a pair of classifiers. Note that the conditions should *guarantee* reusability: it is not enough to have reusability in most problems, the conditions should be such that, for classifier pairs that meet them, $\varepsilon_{cons,AL,n} \leq \varepsilon_{cons,RD,n}$ on *all* problems.

Necessary condition 1: $\mathcal{H}_{cons} \subseteq \mathcal{H}_{sel}$

The first conditions that is necessary to guarantee sample reusability is that the hypothesis space of the consumer is a subset of the hypothesis space of the selector. Suppose that the hypothesis space of the consumer is *not* a subset of the hypothesis space of the selector, so that $\mathcal{H}_{cons} \backslash \mathcal{H}_{sel} \neq \emptyset$.

The active selection is focused on the best hypothesis in $\mathcal{H}_{sel}$, to approximate h^*_{sel} and to make sure that $\text{err}\left(h^*_{sel}, S_{AL,n}\right)$ is better than the $\text{err}\left(h, S_{AL,n}\right)$ of every other hypothesis $h \in \mathcal{H}_{sel}$. With a limited number of samples, more focus on $\text{err}\left(h^*_{sel}\right)$ means less focus on other hypotheses: the active learner undersamples some areas of the sample space. The hypotheses in $\mathcal{H}_{cons} \backslash \mathcal{H}_{sel}$ are completely new. The active sample selection may include some information about these hypotheses, but that is uncertain and there will be problems where little or no information is available. The random sample selection, on the other hand, did not focus on $\mathcal{H}_{sel}$ and therefore did not have to focus less on $\mathcal{H}_{cons}$: it did not undersample any area. In those cases it is likely that the random sample selection provides more information about $\mathcal{H}_{cons}$ than the active sample selection.

This can be a problem in several ways. If the optimal hypothesis h^*_{cons} is one of the new hypotheses, i.e., if $h^*_{cons} \in \mathcal{H}_{cons} \backslash \mathcal{H}_{sel}$ (Figure 13), the active learner has less information to find the best hypothesis in that area than the random sample. It is likely that there are problems where the hypothesis selected by active learning is worse than the hypothesis selected by random sampling, i.e., where $\text{err}\left(h_{cons,AL,n}\right) > \text{err}\left(h_{cons,RD,n}\right)$, which means that $\varepsilon_{cons,AL,n} > \varepsilon_{cons,RD,n}$ and there is no sample reusability.

Even if the optimal hypothesis h^*_{cons} is one of the hypotheses in $\mathcal{H}_{sel}$ – and even if $h^*_{cons} = h^*_{sel}$ – active learning might still select the wrong hypothesis (Figure 14). The active selection may not have sufficient information to see that every new hypothesis in $\mathcal{H}_{cons} \backslash \mathcal{H}_{sel}$ is worse: there are problems where the information in the active selection is such that there is a hypothesis $h \in \left(\mathcal{H}_{cons} \backslash \mathcal{H}_{sel}\right)$ that, on the active sample selection, is better than h^*_{cons}. This might happen to random samples as well, of course, but it is more likely to happen with the active selection. In that case, $\text{err}\left(h_{cons,AL,n}\right) < \text{err}\left(h_{cons,RD,n}\right)$, which implies that $\varepsilon_{cons,AL,n} > \varepsilon_{cons,RD,n}$ and that there is no sample reusability.

Apparently, reusability can not be guaranteed if the consumer can find hypotheses that the selector did not have to consider. There may be reusability in individual cases, but in general, $\mathcal{H}_{cons} \subseteq \mathcal{H}_{sel}$ is a necessary condition for reusability.

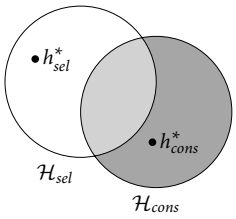


Figure 13: If $h^*_{cons} \notin \mathcal{H}_{sel}$, the active selection may not have enough information to find the optimal hypothesis in $\mathcal{H}_{cons} \backslash \mathcal{H}_{sel}$, the grey area: the active learner focused on h^*_{sel} and did not need information for hypotheses outside $\mathcal{H}_{sel}$.

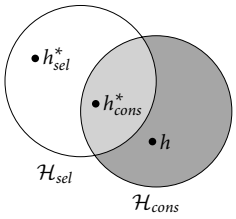


Figure 14: Even if $h^*_{cons} \in \mathcal{H}_{sel}$, the active selection may not have enough information to find h^*_{cons}: there may not be enough information to reject every hypothesis in the grey area. Does it know that h is not optimal?

Necessary condition 2: $h^*_{sel} \in \mathcal{H}_{cons}$

A second condition is that the optimal hypothesis h^*_{sel} for the selector should also be in the hypothesis space of the consumer. Suppose that this is not the case, that $\mathcal{H}_{sel} \supseteq \mathcal{H}_{cons}$, but that $h^*_{sel} \notin \mathcal{H}_{cons}$ (Figure 15).

Then h^*_{cons} was one of the hypotheses that were available to the selector: $h^*_{cons} \in \mathcal{H}_{sel}$. But it is not the optimal solution in $\mathcal{H}_{sel}$, and that may be a problem. There will be enough examples to show that h^*_{sel} was better than h^*_{cons}, since h^*_{cons} was available to the active learner but was not selected. But to be selected as h^*_{cons}, there should be examples in the sample selection that show that the hypothesis is better than any other $h \in \mathcal{H}_{cons}$. There is no guarantee that that information is not available: since it was not a question that the selector needed to answer, the examples that are needed to answer the question may not have been selected. There must be problems where the random sample provides more information near h^*_{cons} than the active selection. In that case it is likely that $h_{cons,RD,n}$ is closer to h^*_{cons} than $h_{cons,AL,n}$. This means that $\text{err}\left(h_{cons,AL,n}\right) > \text{err}\left(h_{cons,RD,n}\right)$, that $\varepsilon_{cons,AL,n} > \varepsilon_{cons,RD,n}$ and that there is no reusability.

Apparently, reusability can not be guaranteed if the consumer finds a different hypothesis than the selector. There may be reusability in individual cases, but in general, $h^*_{sel} \in \mathcal{H}_{cons}$ is a necessary condition for reusability.

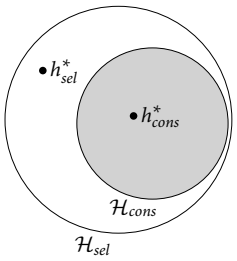


Figure 15: If $h^*_{sel} \notin \mathcal{H}_{cons}$, the selector has enough samples to show that h^*_{sel} is better than any hypothesis in $\mathcal{H}_{sel}$. There may not be enough information to optimise and find h^*_{cons} in $\mathcal{H}_{cons}$.

Sufficient conditions?

The two conditions are necessary to guarantee sample reusability: without $\mathcal{H}_{sel} \supseteq \mathcal{H}_{cons}$ and $h^*_{sel} \in \mathcal{H}_{cons}$ there may be sample reusability in some or even in many problems, but not in all – if there is any reusability, it is due to luck. To guarantee reusability the classifiers need to meet these two conditions, and the conditions are quite strong. The first condition requires that the selector is more powerful than the consumer. The second condition requires that this extra power is not useful: the selector should not find a solution that is better than the solution of the consumer. As a result, the conditions can probably only be met by classifiers that are so similar that they produce the same classifier.

The two necessary conditions do much to improve the chance of reusability, but they are still not sufficient to make a guarantee. The condition $h^*_{sel} \in \mathcal{H}_{cons}$ requires that the selector and the consumer converge to the same hypothesis, but that is only true if there is an infinite sample selection. In practice, reusability should happen at limited sample sizes.

It may be possible to find a condition that guarantees reusability at limited sample size. Here is a condition that can do this – although it may be stronger than absolutely necessary. Consider the situation at a sample size of n samples. The condition $\mathcal{H}_{sel} \supseteq \mathcal{H}_{cons}$ implies that the selector has access to any hypothesis of the consumer. Then the best hypothesis of the consumer has an error on the current sample selection that is at least as large as the error of the best hypothesis of the selector:

$$\text{err}\left(h_{sel,AL,n}, S_{AL,n}\right) \leq \text{err}\left(h_{cons,AL,n}, S_{AL,n}\right)$$
$$\text{err}\left(h_{sel,AL,n}, S_{AL,n}\right) \leq \text{err}\left(h_{cons,AL,n}, S_{AL,n}\right)$$

The importance weights in the sample selection make the error on $S_{AL,n}$ an unbiased estimator for the error on unseen data from the true distribution, i.e., $\mathrm{E}\left[\mathrm{err}\left(h, S_{AL,n}\right)\right] = \mathrm{err}\left(h\right)$, so the previous inequality can be written as

$$\mathrm{err}\left(h_{sel,AL,n}\right) \leq \mathrm{err}\left(h_{cons,AL,n}\right)$$

The same holds for random sampling, so

$$\mathrm{err}\left(h_{sel,RD,n}\right) \leq \mathrm{err}\left(h_{cons,RD,n}\right)$$

Since the active learner is assumed to be functional, the expected error of the classifier selected by self-selection should be better than the expected error with random sampling:

$$\mathrm{err}\left(h_{sel,AL,n}\right) \leq \mathrm{err}\left(h_{sel,RD,n}\right)$$

but there is only reusability if the classifier of the *consumer* is better with active learning than with random sampling, that is, if

$$\mathrm{err}\left(h_{cons,AL,n}\right) \leq \mathrm{err}\left(h_{cons,RD,n}\right)$$

One case where this is *guaranteed* is if the expected errors of the selector and consumer hypotheses are the same. Then

$$\mathrm{err}\left(h_{sel,AL,n}\right) = \mathrm{err}\left(h_{cons,AL,n}\right)$$
$$\mathrm{err}\left(h_{sel,RD,n}\right) = \mathrm{err}\left(h_{cons,RD,n}\right)$$
$$\mathrm{err}\left(h_{cons,AL,n}\right) \leq \mathrm{err}\left(h_{cons,RD,n}\right)$$

This is true if $\mathcal{H}_{cons}$ contains both $h_{sel,AL,n}$ and $h_{sel,RD,n}$.

In other words: the hypothesis space of the consumer should not only contain the optimal hypothesis of the selector, but should also contain any intermediate hypotheses (Figure 16). Reusability can be guaranteed if the consumer can follow the same path towards the solution as the selector.

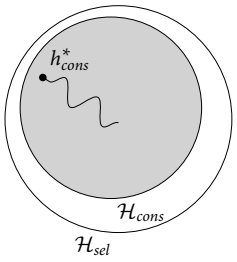


Figure 16: One situation where reusability is guaranteed: $\mathcal{H}_{cons}$ should contain all intermediate hypotheses $h_{sel,AL,n}$ on the way to $h^*_{sel} = h^*_{cons}$.

This result may not be very useful: it says that you can guarantee reusability if the selector and the consumer that are almost or completely the same. That is the definition of self-selection. There is some room between the two required conditions and this sufficient condition, so there might be a less restrictive condition that satisfies $\mathrm{err}\left(h_{cons,AL,n}\right) \leq \mathrm{err}\left(h_{cons,RD,n}\right)$ but does not require the two classifiers to be the same. That, however, might require additional knowledge of the specific classifiers.

An alternative could be to use a more probabilistic approach to the reusability question. Instead of asking for conditions that can guarantee reusability for every possible problem – which, as this section shows, leads to a discussion of the few exceptions where it does not work – it might be enough to find a condition that predicts reusability in most problems, but just not in all.

This would fit in with the proofs of the importance-weighted active learning algorithm itself. Beygelzimer et al. (2010) do not provide absolute guarantees, but give confidence bounds of the performance of their algorithm relative to random sampling. It might be possible to provide similar bounds that predict that, with a certain probability, the performance of importance-weighted active learning with foreign-selection is not much worse than the performance of random sampling. This would probably require strong assumptions about the datasets.

9 *Discussion and conclusion*

Active learning is a wonderful idea. Its promise to deliver better models at a fraction of the cost is very attractive. But active learning has a dark side – in fact, rather many dark sides. The methods that work for the datasets in a study may not work so well on a real dataset, and active learning might give results that are much worse than random sampling. It is hard to evaluate the quality of the sample selection when it is complete, and it is even harder to predict the result before starting the sampling. This unpredictability makes it difficult to justify the use of active learning.

Recent developments in importance-weighted active learning have removed some of the problems. It has a random component, which helps to prevent the missed cluster problem, and it has importance weighting, which helps to remove part of the bias in the sample selection. The behaviour of the algorithm is also described theoretically: there are proven bounds for the sample complexity, label complexity and the expected error of the active learner. This makes it one of the best active learning methods developed so far. The Vowpal Wabbit provides a very efficient implementation. Unfortunately, importance-weighted active learning does not solve every problem.

One of the unsolved questions is that of sample reusability, the topic of this paper. Sample reusability is important in many practical applications of active learning, but it is little-understood. It would be useful if sample reusability could be predicted, but that is a hard problem. The only real study on this topic (Tomanek and Morik, 2011) has inconclusive results. Using uncertainty sampling as the active learning strategy, the study found no pairs of classifiers that always show reusability and did not find a reliable way to predict when reusability would or would not happen. The study examined a number of hypotheses that could have explained the conditions for reusability, but none of these hypotheses were true.

In this paper I investigated the sample reusability in importance-weighted active learning. The authors of the importance-weighted active learning framework suggest that it produces reusable samples (Beygelzimer et al., 2011). This is a reasonable idea: because the algorithm solves many of the problems of uncertainty sampling and the importance weights provide an unbiased sample selection, it might also improve reusability.

However, in this paper I have argued that even importance-weighted active learning does not always produce reusable results (Section 6). If the samples selected for one classifier need to be reusable by *any* other classifier, the active sample selection should be at least as good as the random selection. This is an impossible task, because active learning reduces its label complexity by being worse than random sampling in some areas of the sample space – the areas that do not influence the result. But in general, every sample is interesting to some classifier, so nothing can be left out.

Yes, importance weighting creates an unbiased dataset, so the consumer will always converge to the optimal hypothesis if it is given enough samples (Section 4). This makes the sample selection of importance-weighted active learning potentially more reusable than the sample selection of a selection strategy that can not make this guarantee, such as uncertainty sampling.

But in the practical definition of reusability, active learning should also produce a better hypothesis on a *limited* number of samples, and at limited sample sizes importance weighting does not work that well (Section 5). It corrects the bias on average, so the expected sample distribution is correct, but individual sample selections are still different from the true distribution. The expected error of the hypothesis selected with an active sample can be worse than the expected error of the hypothesis selected with a random sample. Even worse: in some cases, especially at smaller sample sizes, importance weighting introduces so much variability that the results with weighting are worse than the results without.

The results of my practical experiments show that these issues are not only theoretical (Section 7). There is certainly reusability in some cases, for specific combinations of classifiers and datasets. Importance-weighted active learning also seems to produce selections that are more reusable than those of uncertainty sampling – although there are also instances where the opposite is true. However, the many and unpredictable cases where the importance-weighted selections are not reusable make it clear that importance-weighted active learning does not solve the reusability problem.

Are there any certainties in sample reusability? I discussed some of the conditions that might guarantee sample reusability (Section 8). These conditions seem to be quite strong: to get reusability on all possible datasets, the selector and the consumer should be almost exactly the same. For guaranteed reusability, the selector should be able to find all hypotheses of the consumer and the consumer should be able to find the optimal hypothesis of the selector. Even then, if the selector and consumer are not exactly the same, it is possible to find that there is no reusability at smaller sample sizes. This suggests that true universal reusability is impossible.

These are sad conclusions. Importance-weighted active learning may be better than its predecessors, but it does not solve the reusability problem. In fact, it may be impossible for *any* active learner to provide truly reusable results: universal reusability does not exist. Self-selection might work, but foreign-selection is tricky. This makes active learning is an inflexible tool: useful if used for a single task, but an unreliable source for reusable data.

10 Future work

There are some alternative points that I did not discuss in this paper, but that might be worth further study:

- Even if guaranteed reusability on all datasets is impossible, it might still be possible to find conditions that predict reusability in a large number of cases. For most practical applications, there might still be an active learning method that provides reusability on most datasets and for most classifiers. A small reusability risk may be acceptable if the improvements that active learning can offer are important enough. It would be interesting to have a more probabilistic discussion of reusability, instead of the all-or-nothing approach that I used here.

- It might be useful to look at a version of importance-weighted active learning with multiple selectors. In some applications reusability is only needed for a specific set of consumers, and if these consumers are known in advance the selection strategy should be able to create one sample selection that can be shared by all consumers. Of course, an active learner with multiple selectors will require more samples than an active learner with just one selector, but it might still be better than random sampling.

- Quasi-random sampling is not an active learning method, but perhaps it could be used to the same effect. Quasi-random sampling, or low-discrepancy sampling, is a sampling method that has the same expected distribution as random sampling, but with a more even distribution of the samples. A random sample selection has random peaks and dips in its sampling density; quasi-random sampling tries to avoid these peaks and dips and create a more regular sample selection. This may be useful at small sample sizes: the classifier will receive a more balanced sample selection with information from the whole sample space, so it might learn more than it would from a random sample. As the number of samples increases, however, any positive effects will probably soon disappear, but it might improve the beginning.

- Finally, there is a strange situation in active learning where the classifier is always assumed to optimise the zero-one classification error, but in reality may be optimising something else. For example, linear discriminant analysis maximises the likelihood instead of the classification error. If this classifier is then used in an active learning strategy that assumes that it returns the hypothesis that minimises the empirical risk – well, any result is possible. This issue is not specific to sample reusability, but it may be relevant for active learning. Importance-weighted active learning may work with other loss functions than zero-one loss. Further study is needed to show if this is true, and if this is really an issue.

References

Jason Baldridge and Miles Osborne. Active learning and the total cost of annotation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 9–16, 2004.

Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance Weighted Active Learning. *Proceedings of the 26th International Conference on Machine Learning*, 2009.

Alina Beygelzimer, John Langford, Daniel Hsu, and Tong Zhang. Agnostic Active Learning Without Constraints. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

Alina Beygelzimer, Daniel Hsu, Nikos Karampatziakis, John Langford, and Tong Zhang. Efficient Active Learning. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.

Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. In *Proceedings of the 25th International Conference on Machine learning*, 2008.

Wei Fan and Ian Davidson. On Sample Selection Bias and Its Efficient Correction via Model Averaging and Unlabeled Examples. In *SIAM International Conference on Data Mining (SDM'07)*, pages 320–331, 2007.

A. Frank and A. Asuncion. UCI Machine Learning Repository, 2010.

Rong Hu. *Active Learning for Text Classification*. PhD thesis, Dublin Institute of Technology, 2011.

David D. Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine Learning: Proceedings of the Eleventh International Conference*, 1994.

David D. Lewis and William A. Gale. A Sequential Algorithm for Training Text Classifiers. *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12, 1994.

R Development Core Team. R: A Language and Environment for Statistical Computing, 2012.

Julia Schiffner and Stefanie Hillebrand. locClass: Collection of Local Classification Methods, 2010.

Katrin Tomanek. *Resource-Aware Annotation through Active Learning*. PhD thesis, Technische Universität Dortmund, 2010.

Katrin Tomanek and Katharina Morik. Inspecting Sample Reusability for Active Learning. In *JMLR: Workshop and Conference Proceedings 16 (2011): Workshop on Active Learning and Experimental Design*, 2011.

Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. *Proceedings of the 21st international conference on Machine learning*, page 114, 2004.

Working document

Working document

Gijs van Tulder

22 October 2012

1 First proposal

Sample reusability is an important problem in active learning. In the first systematic experiments in this area, Tomanek and Morik (2011) evaluate several hypotheses of reusability. However, they only use the simplest sample selection strategy: uncertainty sampling, which is very aggressive and introduces a strong sample selection bias. Since sample reusability and sampling bias are related, one would expect that less aggressive strategies have a better sample reusability.

I would like to do two experiments:

1. Experiment with uncertainty sampling and several classifiers, to replicate the reusability experiments done by Tomanek and Morik (2011).
2. Experiment with other, less aggressive sample selection strategies, to see if aggressiveness has an effect on sample reusability.

Tomanek and Morik (2011) used five datasets from the UCI Machine Learning Repository[1]: CAR, MUSHROOM, NURSERY, SEGMENT and SICK. They also did experiments with Named Entity Recognition, but I think it is better to concentrate on the 'simple' classification problems first.

[1] http://archive.ics.uci.edu/ml/

For the reusability experiments it is necessary to select a sample selection strategy that works with more than one classifier. This is a real problem: while simple strategies such as uncertainty sampling can be used with almost any classifier, the mellow algorithms are often more advanced and more tailored to one type of classifier. For example, the A^2 algorithm maintains a large candidate set of hypotheses, which may be hard to implement with a support vector machine.

The algorithm discussed by Beygelzimer et al. (2010) seems to be the most promising.[2] The algorithm is based on the A^2 and Importance Weighted Active Learning (IWAL) algorithms. While the early algorithms use a set of candidate hypotheses, Beygelzimer et al. use an oracle that returns an empirical risk minimizing (ERM) hypothesis. Since many supervised learning algorithms produce an ERM hypothesis, Beygelzimer et al. write that their algorithm is "immediately and widely applicable".

[2] See "The End of the Beginning of Active Learning", a blog post by Daniel Hsu and John Langford in April 2011. http://hunch.net/?p=1800

Tomanek and Morik (2011) use the REU score to quantify reusability:

$$\mathrm{REU}\left(S_{\mathrm{frgn}}, S_{\mathrm{self}}, S_{\mathrm{base}}, a, b\right) = \frac{\mathrm{AUC}\left(S_{\mathrm{frgn}}, a, b\right) - \mathrm{AUC}\left(S_{\mathrm{base}}, a, b\right)}{\mathrm{AUC}\left(S_{\mathrm{self}}, a, b\right) - \mathrm{AUC}\left(S_{\mathrm{base}}, a, b\right)} - 1$$

With uncertainty sampling as the selection strategy, I tried to replicate the results of Tomanek and Morik. I ran experiments with three datasets from the UCI repository and calculated the REU scores for combinations of classifiers. The results (see table 1 where not similar to those of Tomanek and Morik: the range of REU scores was different, as were the signs.

CAR	naivebc	libsvc	loglc
naivebc	0	-0.2151346748	-0.8955995989
libsvc	-1.1335044723	0	0.6865867087
loglc	-0.6824149121	0.1931244485	0
MUSHROOM	naivebc	libsvc	loglc
naivebc	0	-6.1031565612	-7.22489062
libsvc	-1.111858329	0	-0.3731496208
loglc	0.1587352218	-0.7526162947	0
NURSERY	naivebc	libsvc	loglc
naivebc	0	-2.3361888978	-0.2722452972
libsvc	-134.3928094887	0	-0.7268834511
loglc	-44.4630226275	-4.5191760879	0

Table 1: Reusability results, for comparison with the results of Tomanek and Morik (2011). REU greater than 0 is strong reusability (foreign-selection is better than self-selection). REU between 0 and -1 is reusability (foreign-selection better than random sampling). REU less than -1 is no reusability (foreign-selection worse than random sampling).

2 *Experiments with AALWC*

I implemented the 'agnostic active learning without constraints' (AALWC) algorithm by Beygelzimer et al. (2010).

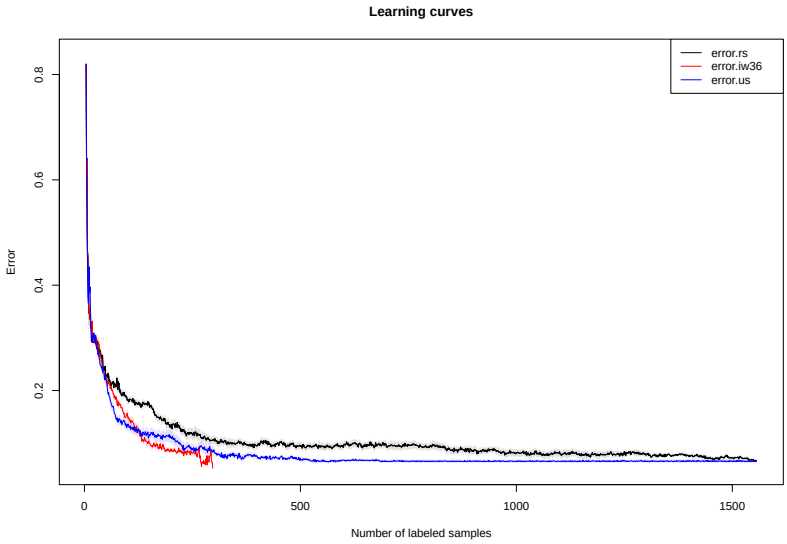


Figure 1: Experiments with the CAR dataset. Importance-weighted SVM for AALWC (IW36, $C_0 = 36$) and uncertainty sampling (US) against random sampling (RS).

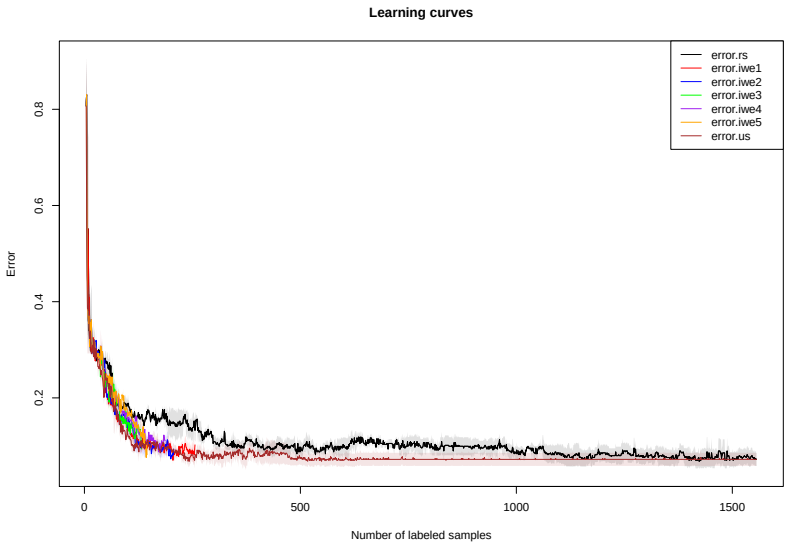
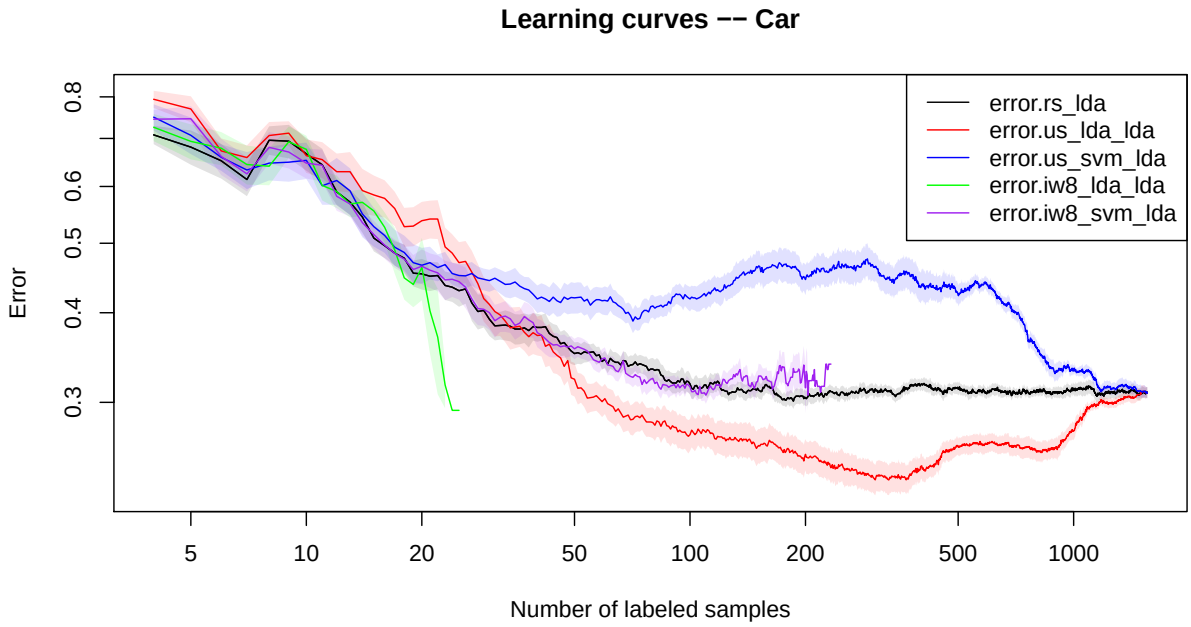


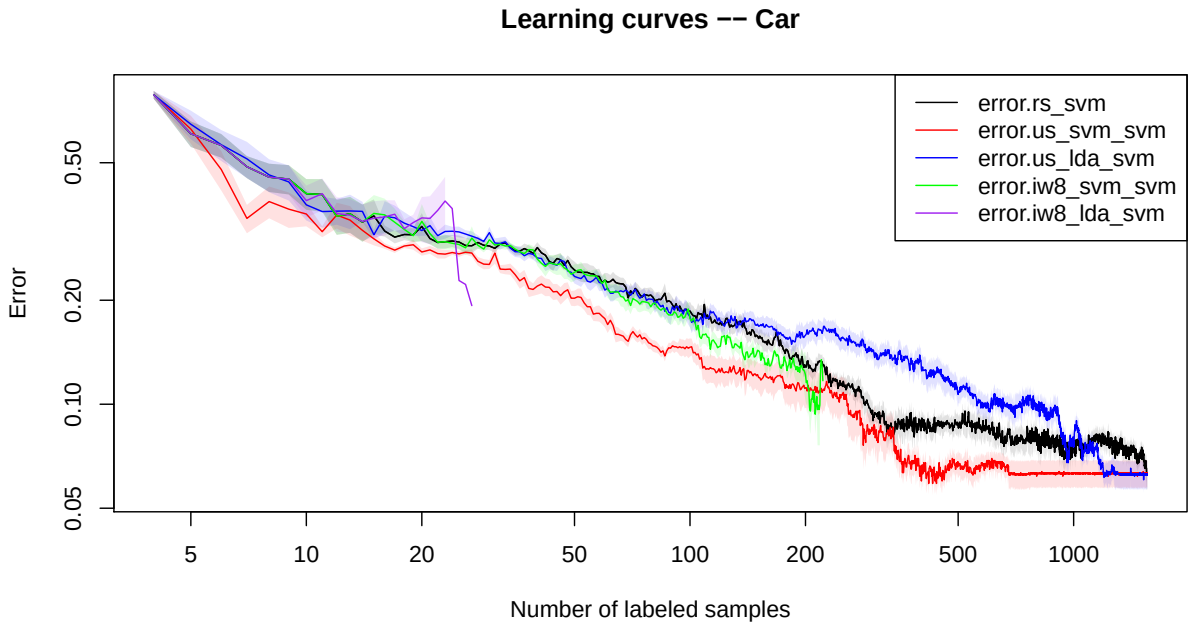
Figure 2: Experiments with the CAR dataset. Importance-weighted SVM for AALWC, comparing different values for the C_0 parameter ($C_0 = 10^{\{-1,-2,-3,-4,-5\}}$).

3 *Experiments with AALWC*

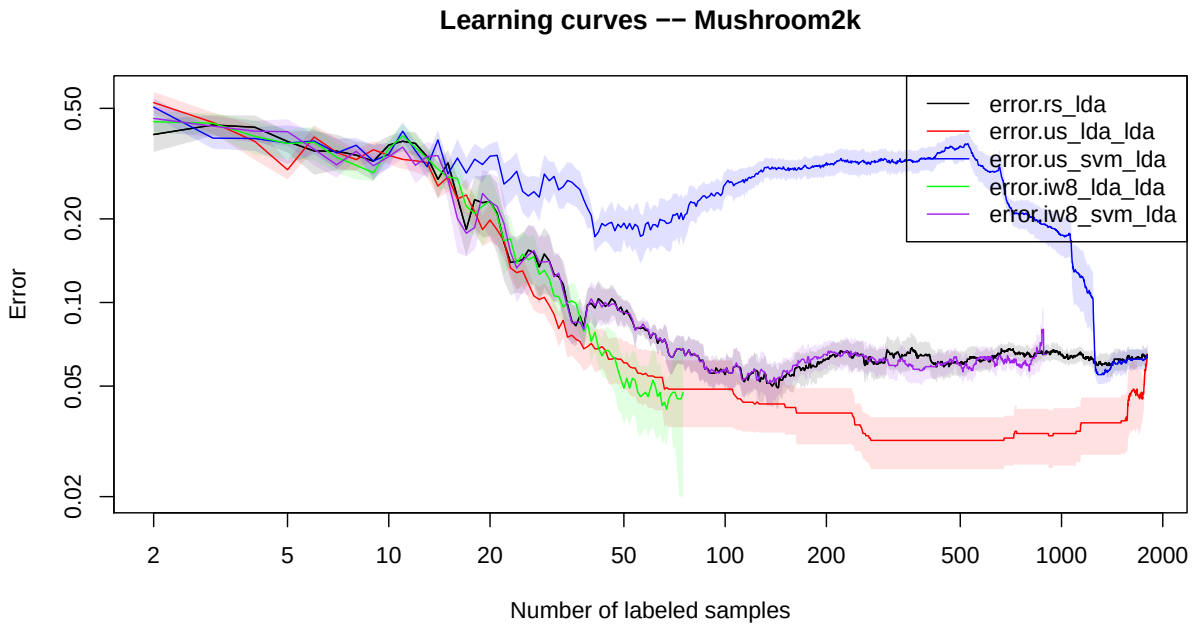
Reusability experiments with US and AALWC experiments, with importance-weighted LDA and SVM classifiers.

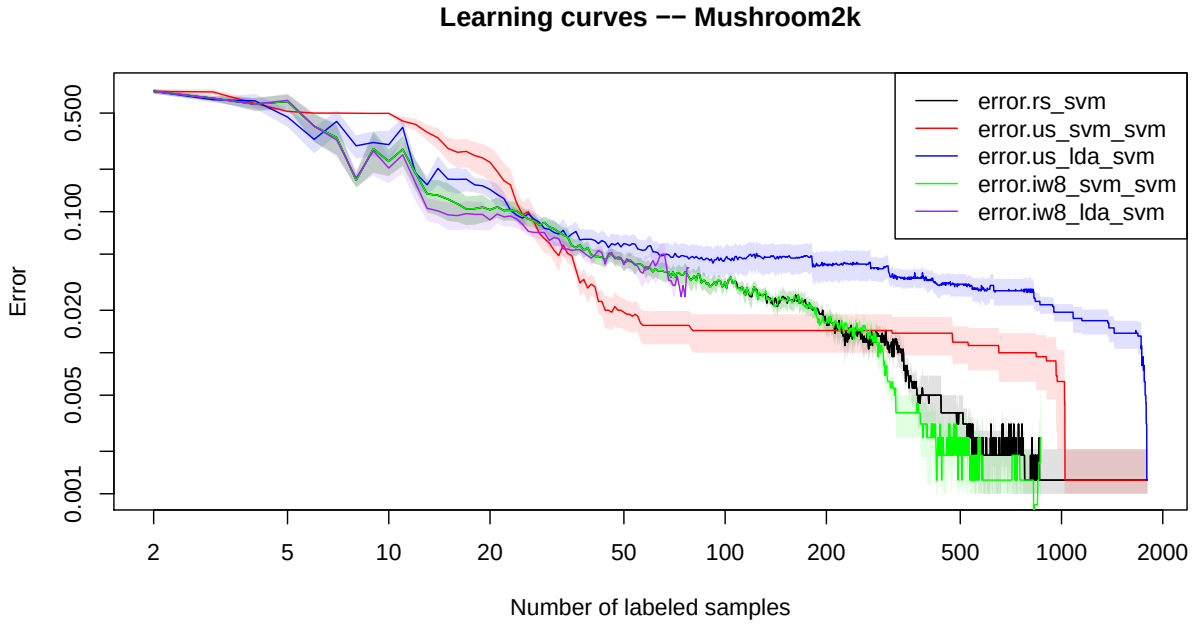
Reusability experiments with the UCI CAR dataset, comparing random sampling, uncertainty sampling and AALWC (IW8, $C_0 = 8$). Top: IW-LDA consumer, IW-LDA/IW-SVM selectors. Bottom: IW-SVM consumer, IW-LDA/IW-SVM selectors.



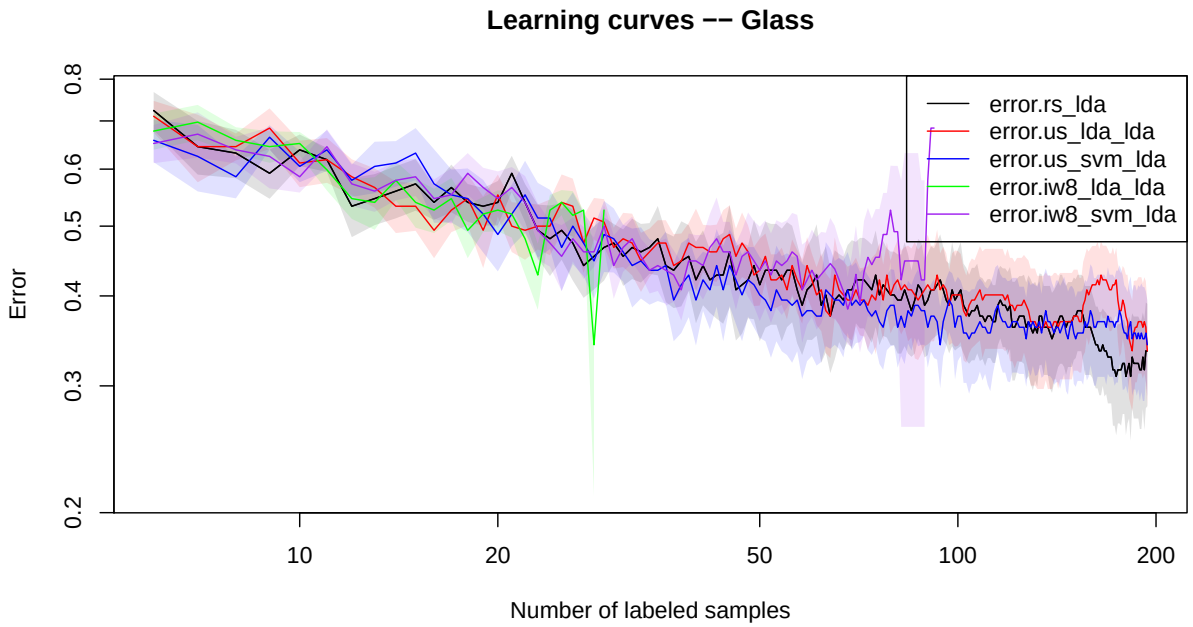


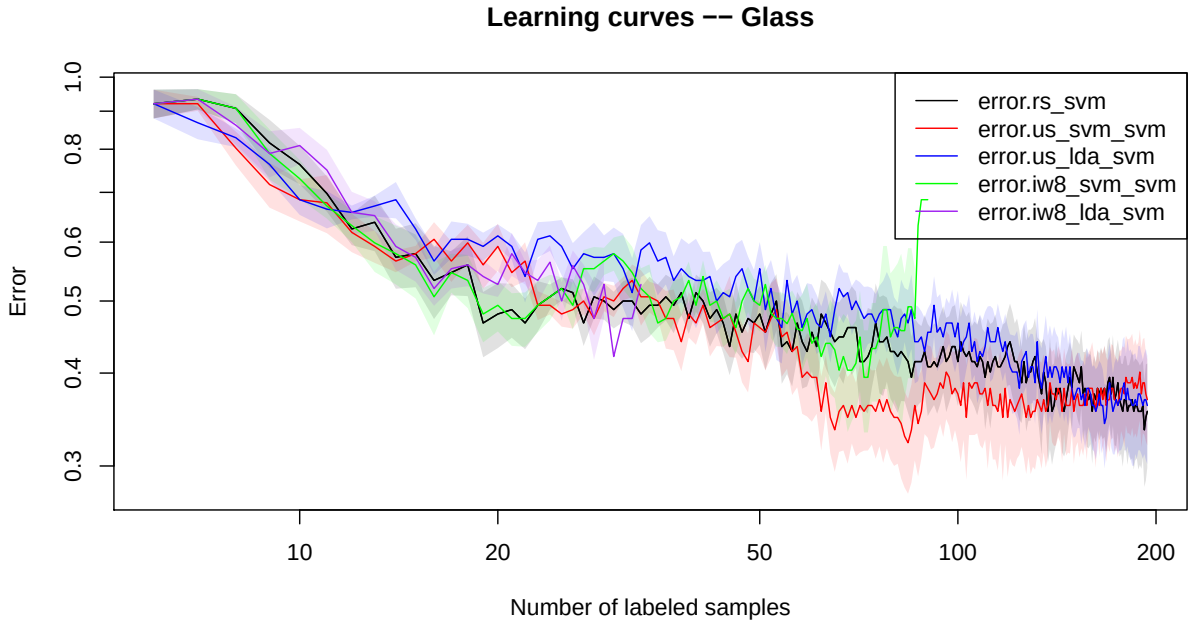
Reusability experiments with a subset of the UCI MUSHROOM dataset, comparing random sampling, uncertainty sampling and AALWC (IW8, $C_0 = 8$). Top: IW-LDA consumer, IW-LDA/IW-SVM selectors. Bottom: IW-SVM consumer, IW-LDA/IW-SVM selectors.



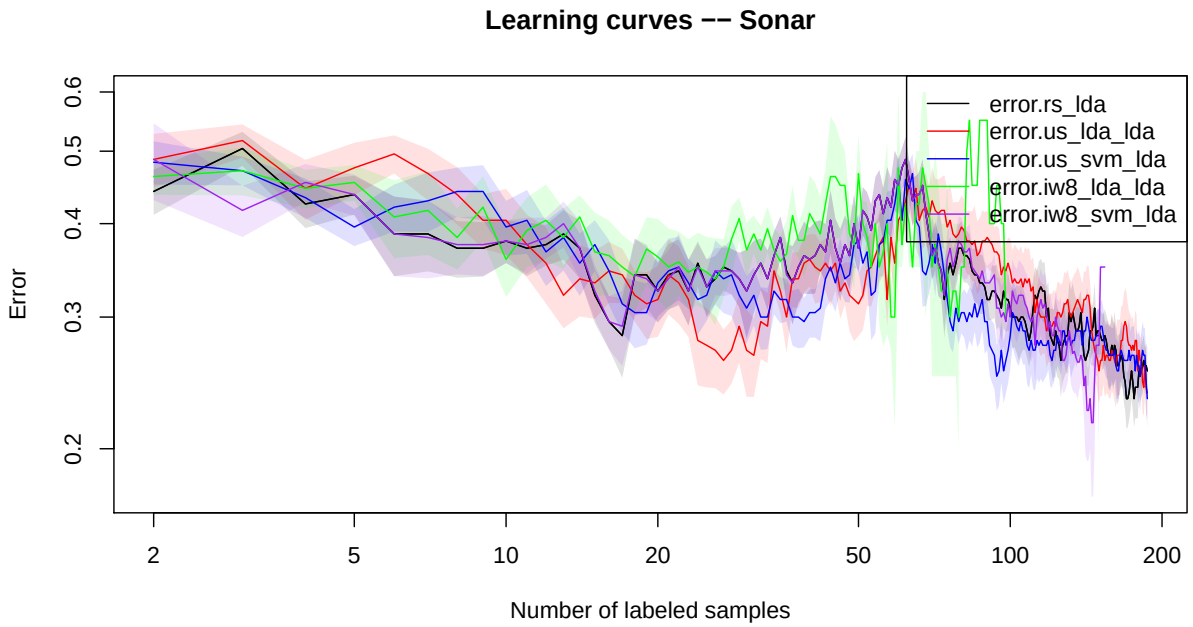


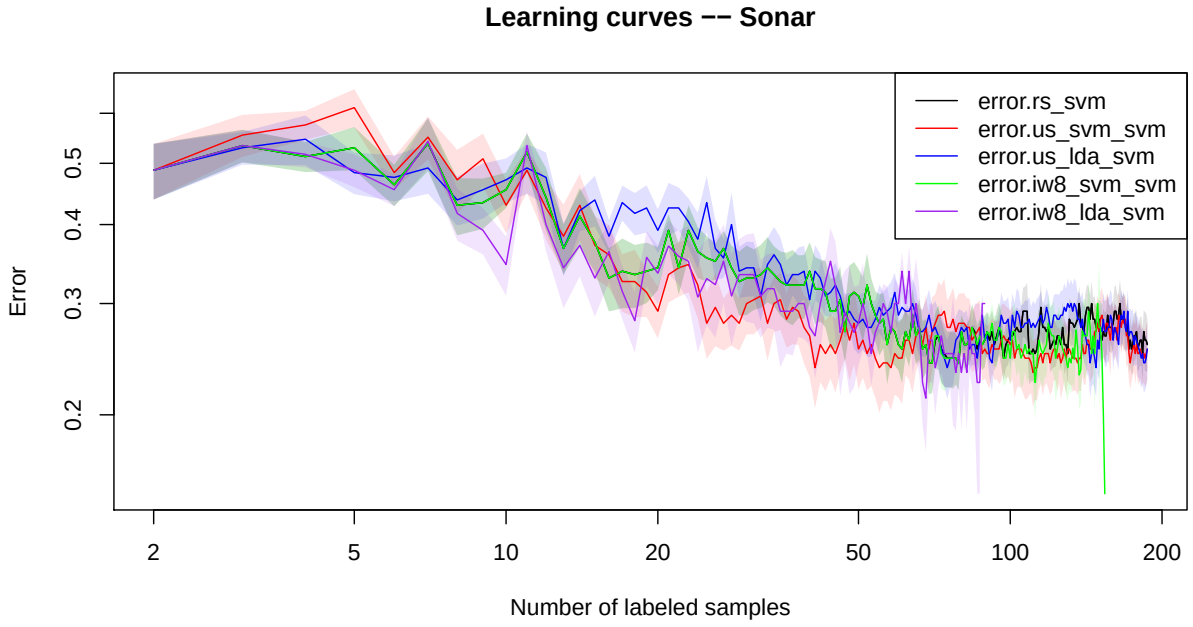
Reusability experiments with a subset of the UCI GLASS dataset, comparing random sampling, uncertainty sampling and AALWC (IW8, $C_0 = 8$).
Top: IW-LDA consumer, IW-LDA/IW-SVM selectors. Bottom: IW-SVM consumer, IW-LDA/IW-SVM selectors.





Reusability experiments with a subset of the UCI SONAR dataset, comparing random sampling, uncertainty sampling and AALWC (IW8, $C_0 = 8$). Top: IW-LDA consumer, IW-LDA/IW-SVM selectors. Bottom: IW-SVM consumer, IW-LDA/IW-SVM selectors.





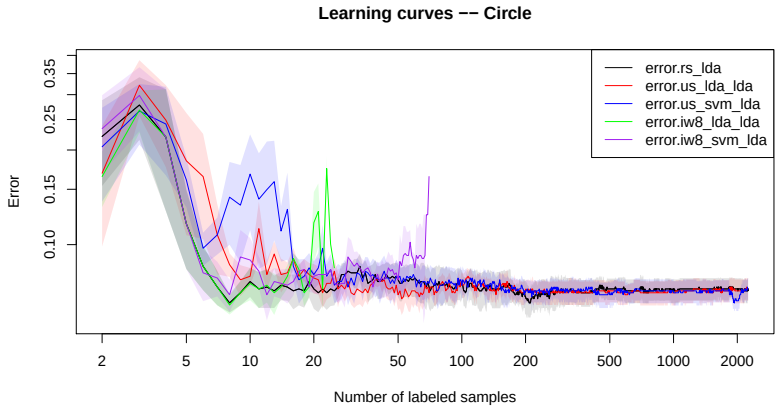
4 *Experiments with AALWC*

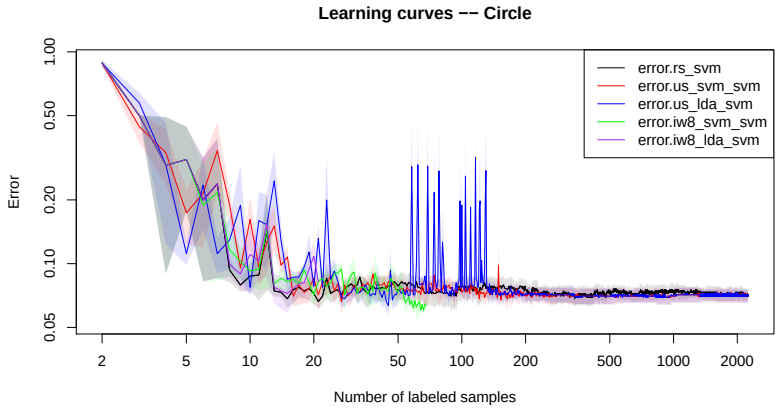
I fixed a few bugs in the importance-weighted LDA. I have looked for an incremental LDA, too speed up the experiments, but have not yet been able to implement it. (The mean and covariance matrices can be calculated incrementally, but that is not enough.)

With the new code, I did a number of experiments, see results on the next pages.

Reusability experiments with a synthetic dataset of two 2D-gaussians, comparing random sampling, uncertainty sampling and AALWC (IW8, $C_0 = 8$). Top: IW-LDA consumer, IW-LDA/IW-SVM selectors. Bottom: IW-SVM consumer, IW-LDA/IW-SVM selectors.

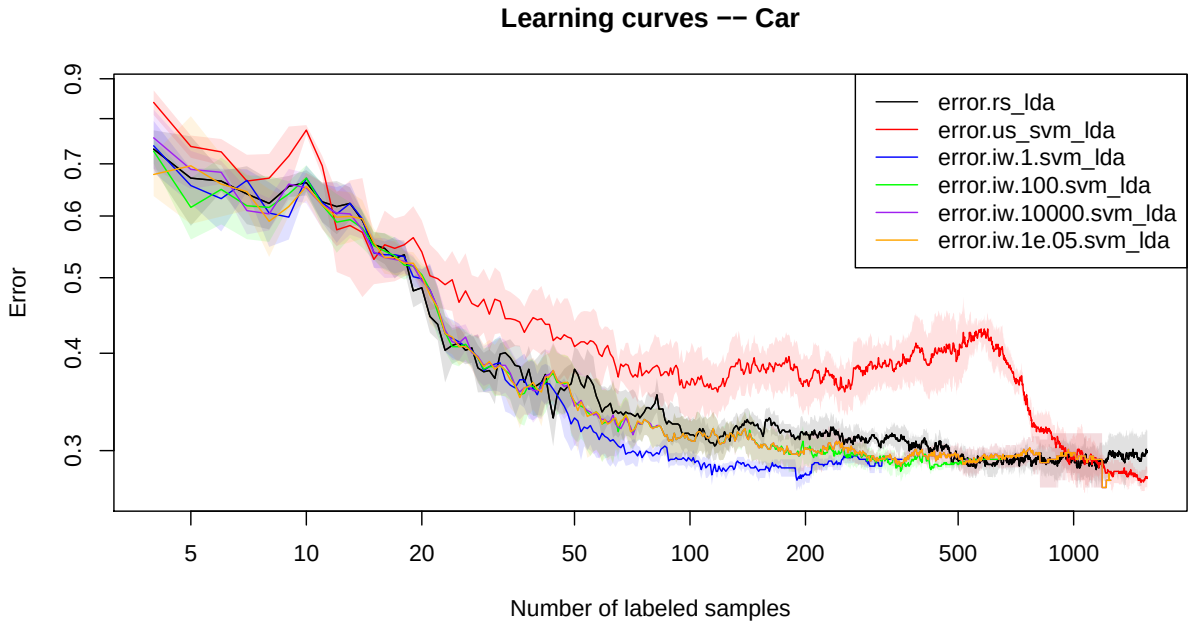
Apparently this problem is too simple.

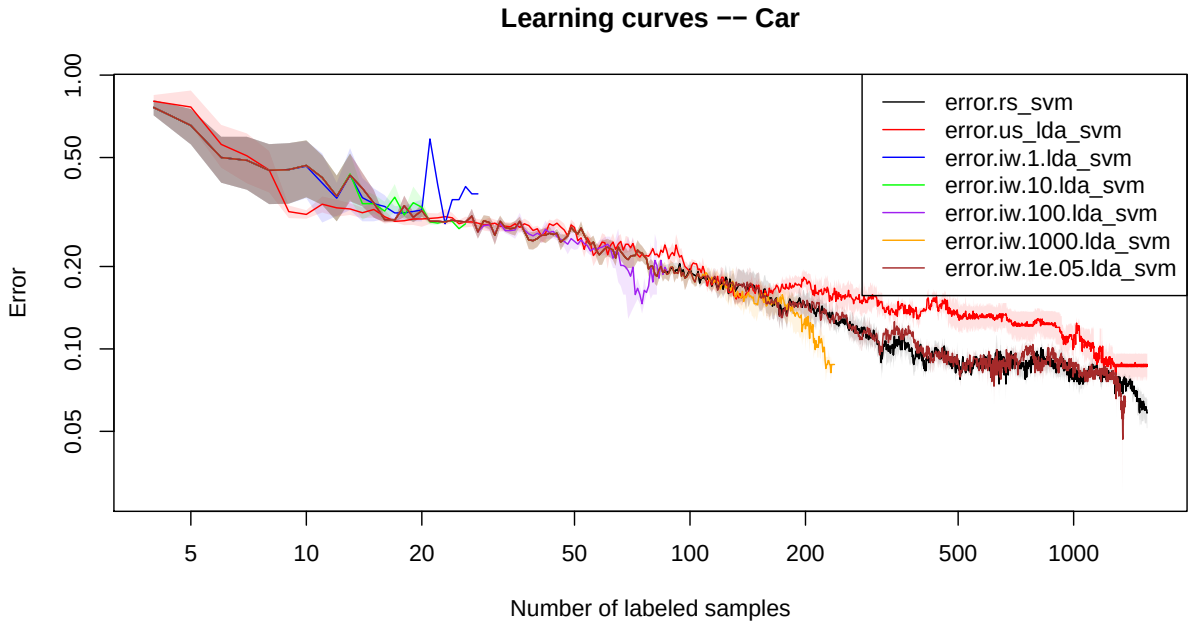




Reusability experiments with a subset of the UCI CAR dataset, comparing different values for the C_0 parameter. ($C_0 = 1, 100, 10000, 10.000$)). Top: IW-SVM selector, IW-LDA consumer (only foreign selection). Bottom: IW-LDA selector, IW-SVM consumer (only foreign selection).

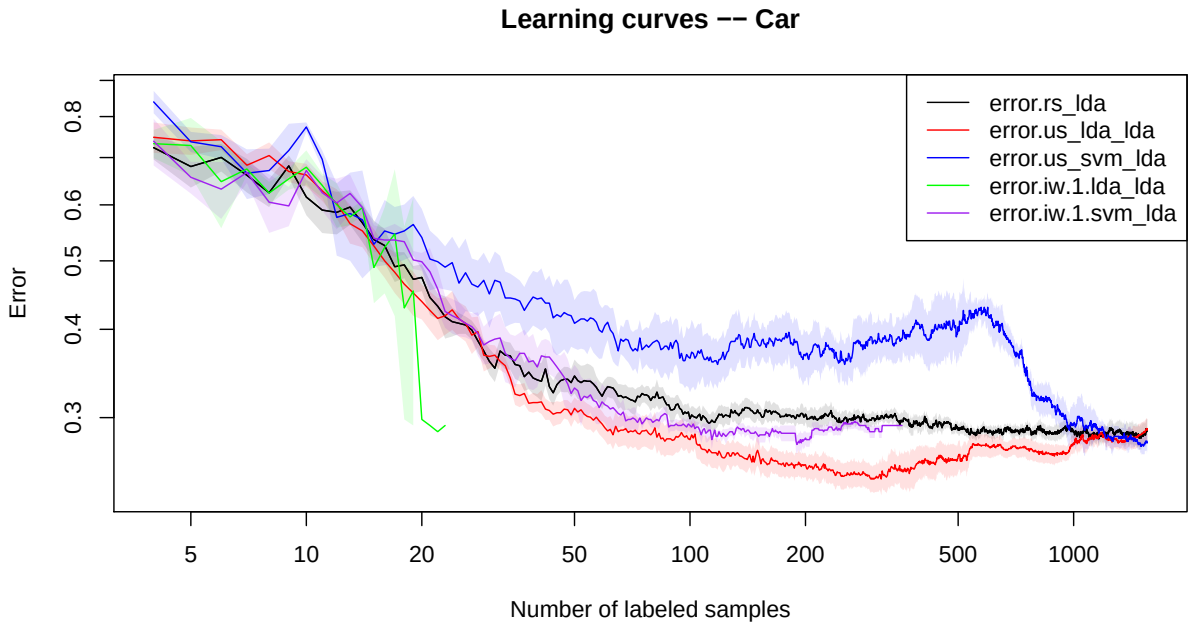
A larger C_0 corresponds to a less aggressive algorithm. The graphs show that the less aggressive learners select more examples than the more aggressive learners (the lines are longer).

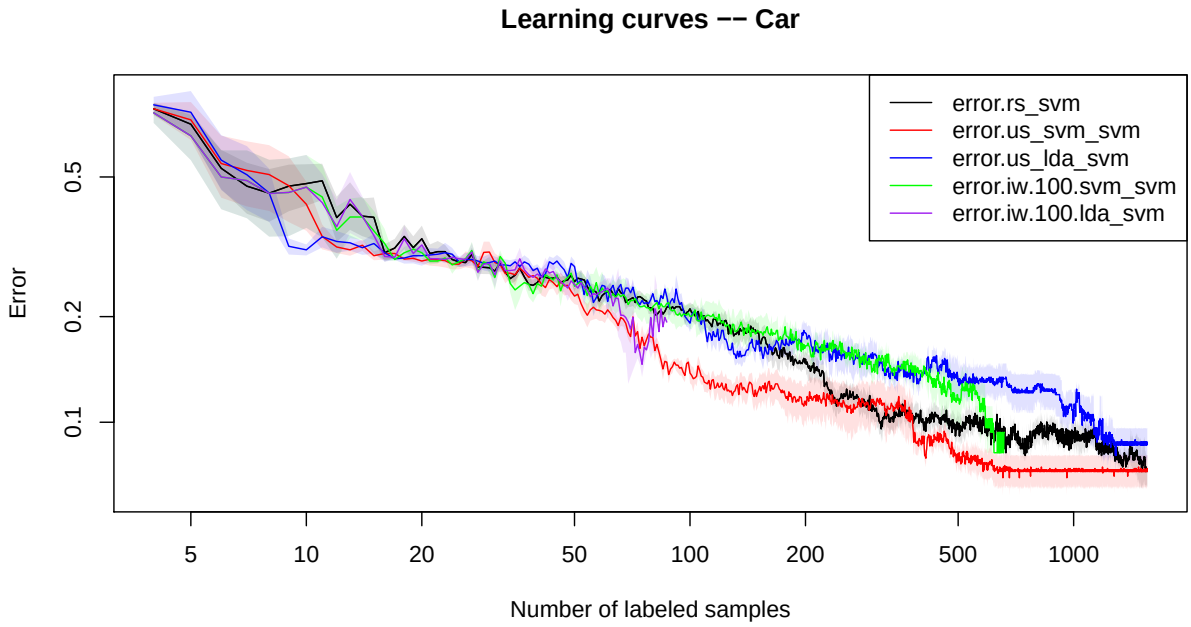




Reusability experiments with a subset of the UCI CAR dataset, comparing foreign and self-selection. Top: IW-SVM selector, IW-LDA/SVM consumer. Bottom: IW-LDA selector, IW-LDA/SVM consumer.

AALWC foreign-selection by SVM for LDA works quite well; better than random sampling and not much worse than self-selection. The self-selecting SVM with AALWC performs worse than foreign-selecting LDA with AALWC.





5 *Experiments plus intuition about AALWC*

First some notes about 'agnostic active learning without constraints' (AALWC) by Beygelzimer et al. (2010). Paul Mineiro[3] has a few useful notes that I read before I wrote the following.

[3] "Agnostic Active Learning Without Constraints, Explained", a blog post by Paul Mineiro at `http://www.machinedlearnings.com/2012/01/agnostic-active-learning-without.html`

Definitions

The algorithm is a sequential active learner: in iteration k the algorithm looks at x_k, the k'th example. Define S_k as the set of importance-weighted samples selected before iteration k. The algorithm will query the label of item x_k with a certain probability, P_k.

The true optimal hypothesis h^* minimises the empirical risk:

$$h^* = \arg\min_{h \in \mathcal{H}} \operatorname{err}(h)$$

The current hypothesis h_k minimises the importance-weighted empirical risk given S_k, the collection of importance-weighted samples collected up to iteration k:

$$h_k = \arg\min_{h \in \mathcal{H}} \operatorname{err}(h, S_k)$$

The current hypothesis will have a prediction of the label of the new example x_k: $h_k(x_k)$. Define an alternative hypothesis h'_k, which is the risk-minimising hypothesis from the set of hypotheses that disagree with h_k about the label of x_k:

$$h'_k = \arg\min_{h \in \mathcal{H} \wedge h(x_k) \neq h_k(x_k)} \operatorname{err}(h, S_k)$$

Sampling

While h^* is unknown, there are only two possibilities: the current hypothesis h_k disagrees with the optimal hypothesis h^* on the label of x_k, or the hypotheses agree (though the label may be incorrect).

Suppose that the current hypothesis disagrees with optimal hypothesis, i.e. $h_k(x_k) \neq h^*(x_k)$. Then the error of the alternative hypothesis h'_k must be close to the error of the optimal hypothesis h^*, since h'_k is the risk-minimising hypothesis from the set of hypotheses that includes that includes the optimal hypothesis h^*. This means that $\operatorname{err}(h'_k, S_k) - \operatorname{err}(h_k, S_k)$ is small (with a certain probability).

On the other hand, the optimal hypothesis may agree with the current hypothesis on x_k. Then the error of the alternative hypothesis h'_k must be larger than the error of the optimal hypothesis h^*. At the same time, the error of the current hypothesis is closer to that of the optimal hypothesis and therefore smaller than that of h'_k. This means that $\operatorname{err}(h'_k, S_k) - \operatorname{err}(h_k, S_k)$ is large (with a certain probability).

The trick of the algorithm is that the selection probability P_k depends on the difference of the errors. If $\operatorname{err}(h'_k, S_k) - \operatorname{err}(h_k, S_k)$ is small this implies that the current hypothesis is wrong about the label of x_k: it is important to sample x_k, so P_k should be large. On the other hand, if $\operatorname{err}(h'_k, S_k) - \operatorname{err}(h_k, S_k)$ is large the current hypothesis is probably correct on x_k: it is not necessary to sample x_k, so P_k should be small.

This is exactly what AALWC does: it queries x_k with probability

$$P_k = \min\left\{1, \left(\frac{1}{G_k^2} + \frac{1}{G_k}\right) \cdot \frac{C_0 \log k}{k-1}\right\} \quad \text{where} \quad G_k = \text{err}\left(h'_k, S_k\right) - \text{err}\left(h_k, S_k\right)$$

If selected, the example is labelled and stored with importance weight $1/P_k$.

Experiments

I did experiments with three classifiers: LDA and SVM with linear and radial kernels. I have tried to visualise the selection order of the uncertainty sampling and AALWC active learning methods for each of these classifiers. In the experiments I used the UCI 'car' dataset I used before, plus two synthetic datasets that I generated.

The first synthetic dataset I called 'mound': a 2D dataset with two classes (figure 3). Class one has a uniform distribution from $x = [-1, 0]$ and $y = [-1, 1]$. Class two has a uniform distribution from $x = [-1, 0]$ and $y = [-1, 1]$. I took the examples in an area defined by a quadratic function from class two and added them to class one. The result is a dataset that can be learned by a quadratic classifier and is slightly more difficult for a linear classifier.

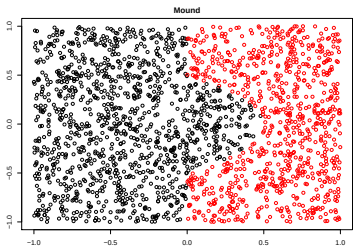


Figure 3: The 'mound' dataset.

'Ex1d, my second dataset, is a 1D two-class dataset (figure 4) that I hoped would confuse the linear classifiers. The dataset has two possible decision boundaries, one at 1.0 and one at 3.0. 80% of the samples are to the left and right of 1.0, with a wide margin. 20% of the samples are to the left and right of 3.0, without a margin. I had hoped that uncertainty sampling, which has a preference for items close to a decision boundary, would make the learner more interested in the lower density area on the right (where everything is close to the decision boundary) instead of the area on the left, which has a higher density area but also a wide margin.

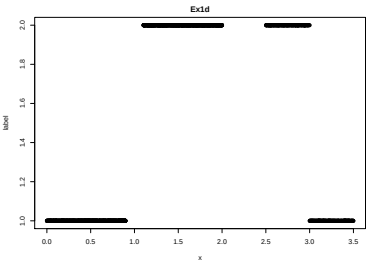


Figure 4: The 'ex1d' dataset. Samples distributed 40%, 40%, 10%, 10%.

In the experiments I looked at the order in which the active learner selected the examples. By plotting the average position of an example for one classifier against the average position in another classifier, it becomes clear how preferences of the classifiers differ.

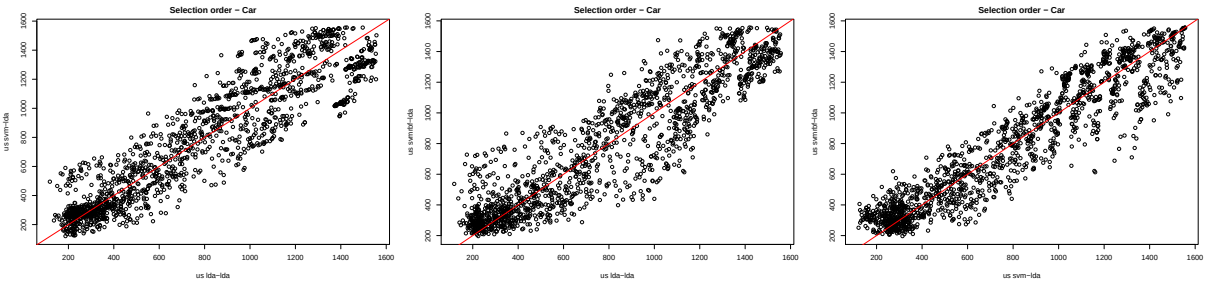


Figure 5: The selection order of samples in the 'car' dataset. With uncertainty sampling, all classifiers select samples in the same order.

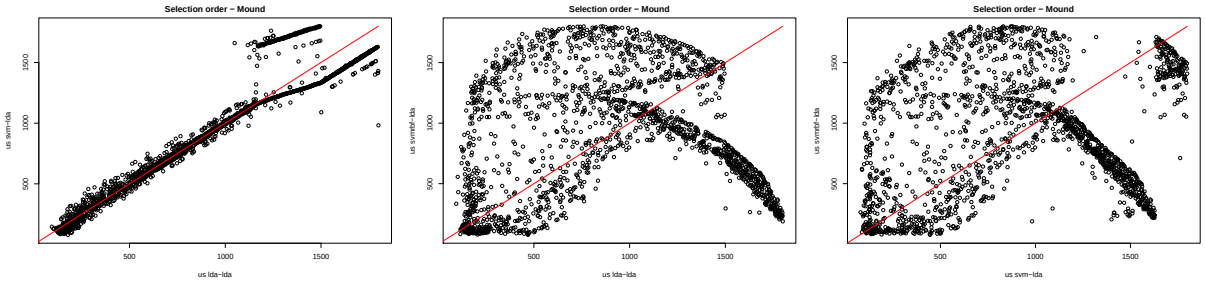


Figure 6: The uncertainty sampling selection order of samples in the 'mound' dataset. LDA and SVM-linear have similar preferences (left), but differ from SVM-radial (middle, right).

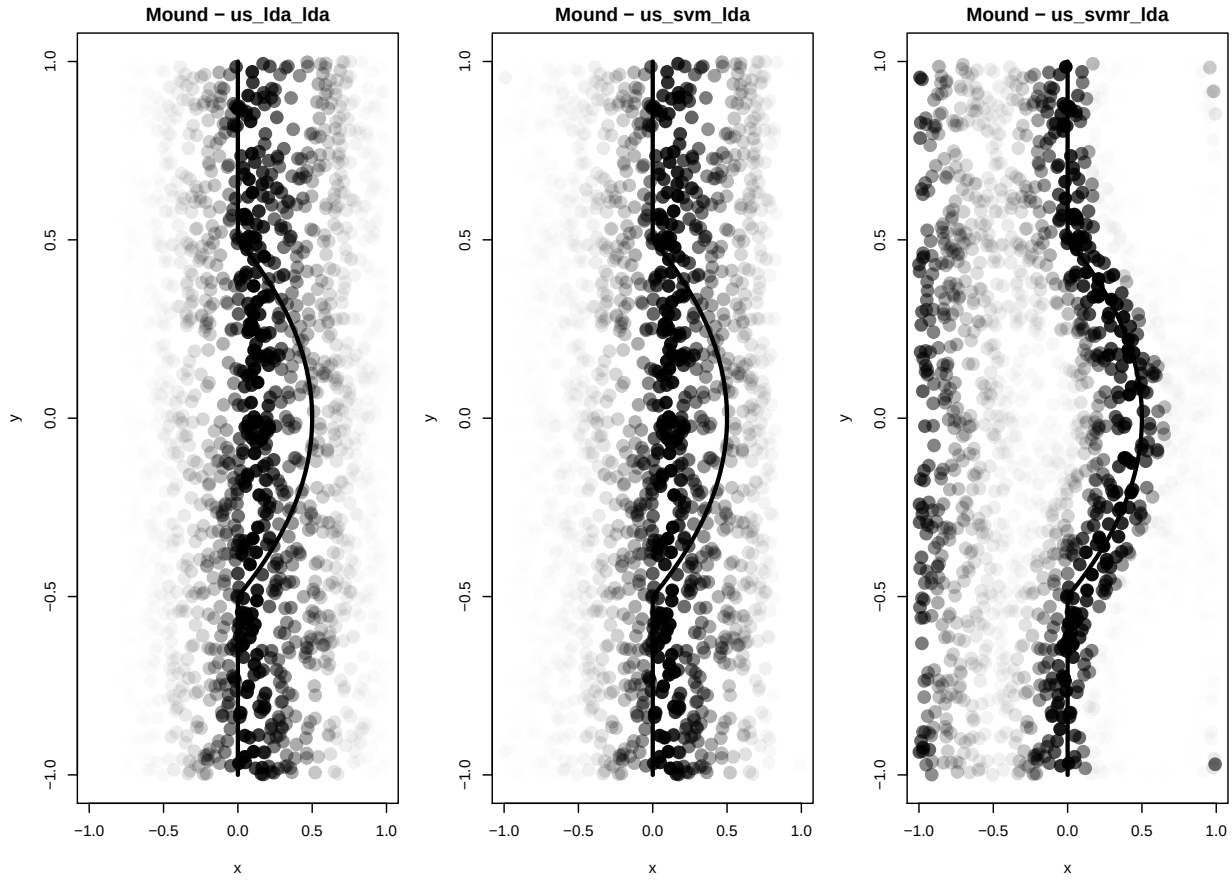


Figure 7: Plotting the selection order of each example shows the different structure of the classifiers. (A darker color means the example was selected earlier.)

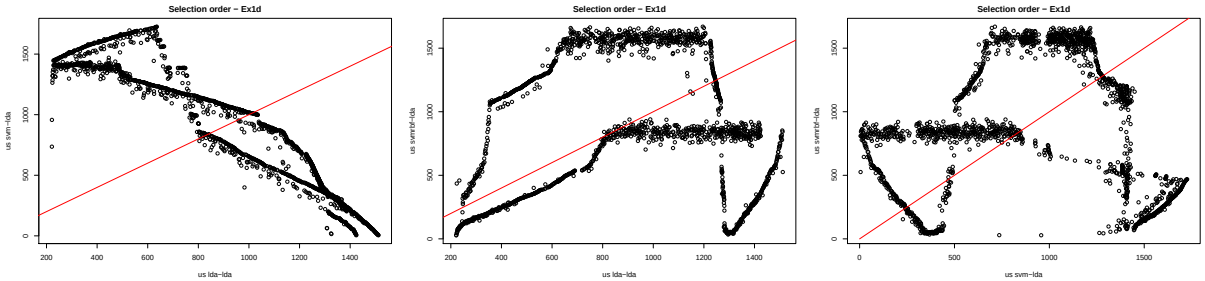


Figure 8: The uncertainty sampling selection order of samples in the 'ex1d' dataset. Different order for each classifier.

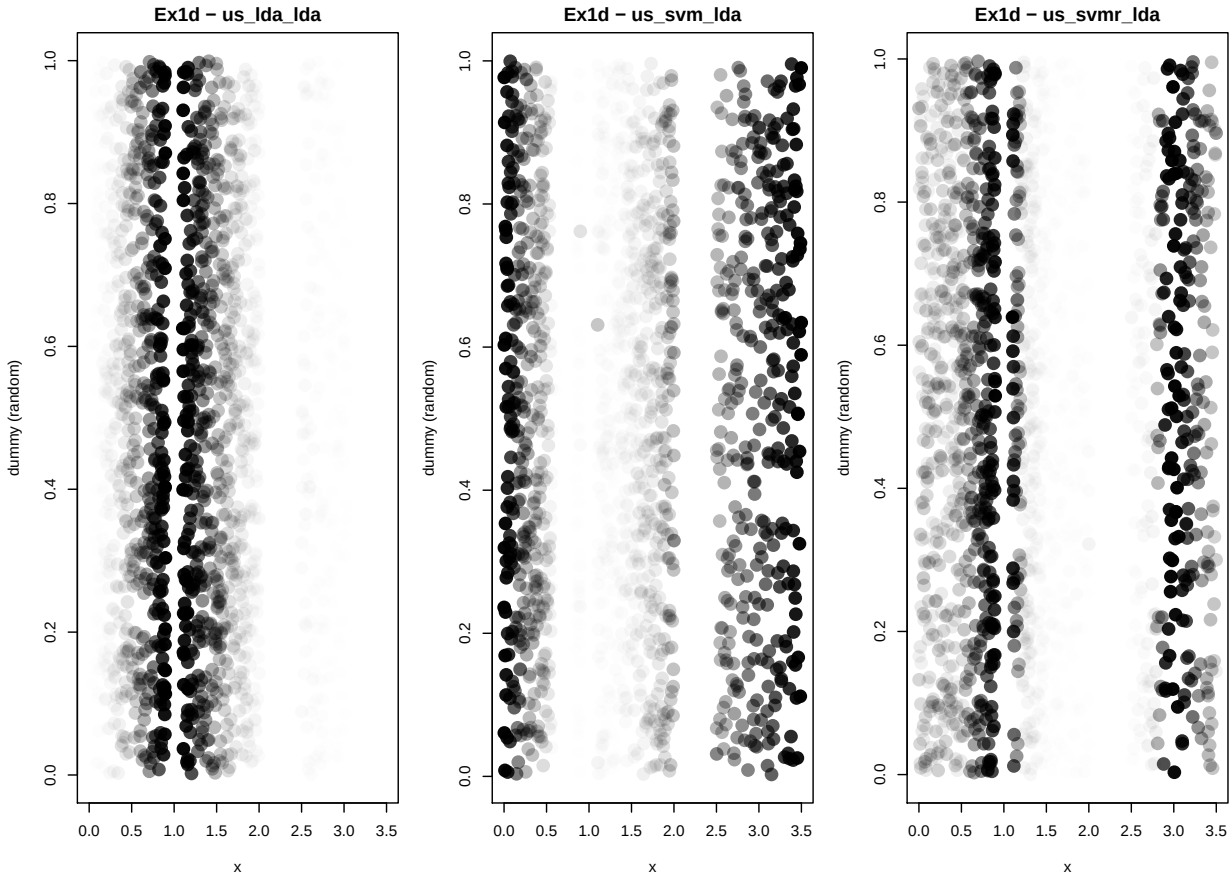


Figure 9: Plotting the selection order of each example shows a interesting pattern. (A darker color means the example was selected earlier.)

6 *Dissecting 'aggressiveness'*

What are the properties of a good active learner? (I.e. dissecting 'aggressiveness'[4].)

An active learning algorithm selects new examples according to a utility function $U(x)$. The utility of an example is the expected reduction in the error of the classifier on the real dataset if that example would be labeled.

[4] The word 'aggressiveness' whas introduced by Dasgupta (2011), who does not give an exact definition of these terms, but writes that "roughly, an aggressive scheme is one that seeks out highly informative query points while a mellow scheme queries any point that is at all informative, in the sense that its label cannot be inferred from the data already seen".

Property 1: aggressiveness

An aggressive algorithm selects the examples with the highest estimated utility first. A less agressive (more mellow) algorithm may look at the estimated utility, but also considers other circumstances in its sample selection.

In theory, if the learner knows $U(x)$, the most aggressive algorithm produces the best result with the lowest sample complexity.

In practice the exact utility $U(x)$ is not known. The active learner makes an estimate $\hat{U}(x)$, based on the information that is available about the labeled and unlabeled examples in the training set.

This estimate $\hat{U}(x)$ can produce several dangers for aggressive active learning algorithms:

1. The estimates may be incorrect. If for examples x_1 and x_2 the estimates $\hat{U}(x_1) > \hat{U}(x_2)$ while the true estimates $U(x_1) < U(x_2)$, the algorithm will not select the best example. It might select examples that are less informative, or not informative at all.

2. The estimates may be so incorrect and so dependent on the initial samples that, with an unlucky initial set, the incorrect estimates can lead to a 'tunnel vision'. Important areas of the sample space are not queried simply because the algorithm does not know that they are important. (The missed cluster effect.)

3. The utility functions of two classifiers may be different. Even if $\hat{U}(x)$ follows the true $U(x)$ for classifier A, it may not be correct for the $U'(x)$ of classifier B. The selection made with $\hat{U}(x)$ is biased towards classifier A. (The sample reusability problem.)

Property 2?

Aggressiveness is both good and evil: we want a very aggressive algorithm because we want a low label complexity, but we do not want the problems that may come with it. It seems therefore that this is not a one-dimensional problem, but that an active learner can be rated on at least two dimensions: its aggressiveness, and something else.

What should be on this second axis? One option would be to look at the 'statistical consistency' of the algorithm. A statistically consistent algorithm will eventually, given unlimited samples, converge to the 'true' optimal solution that minimises the error on the true dataset. The missed cluster effect illustrates what happens if an algorithm is not statistically consistent. Being statistically consistent is important, but it may not cover all problems and it is very restrictive.

A second, maybe better option is the lack of 'agnosticism' of the algorithm, or the strength of the assumptions it makes. All three problems listed in the previous section derive from invalid assumptions:

1. The assumption that the estimates are more or less correct.
2. The assumption that the initial sample selection provides enough information about all structures in the dataset.
3. The assumption that two different classifiers share the same utility functions.

The best active learning algorithm must be an algorithm that is very aggressive, yet does not make too many assumptions. However, such an algorithm may not exist: they are either non-aggressive, or they are aggressive *because they make these assumptions.*

Perhaps what we actually want is an active learning algorithm that is aggressive without making too many *harmful* assumptions.

Experiments

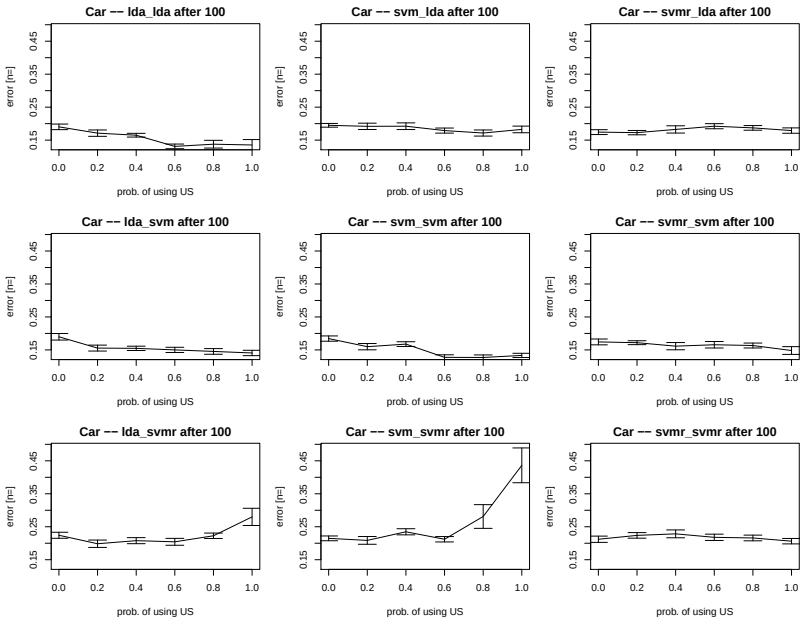


Figure 10: Errors of active learners against increasing aggressiveness, after selecting 100 examples. Before selecting a new example, the algorithm made a random choice between random sampling and uncertainty sampling. The horizontal axis shows the probability of using uncertainty sampling ($p = 0$ is pure random sampling, $p = 1$ is pure uncertainty sampling).

A downwards-sloped line indicates that the uncertainty sampling is on the right track: the more aggressive the algorithm, the better it performs. The 'bump' at the end of some of the graphs indicates reusability problems: too aggressive is not good.

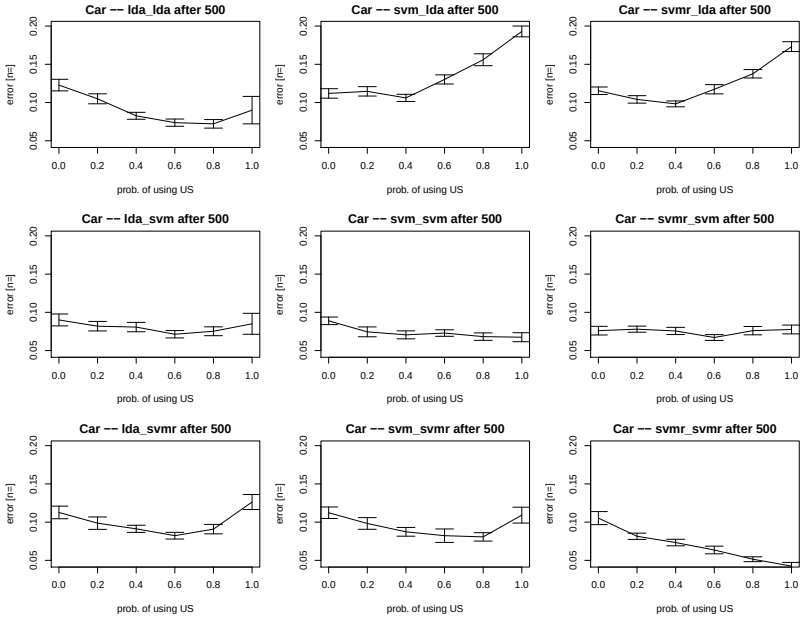


Figure 11: Errors of active learners against increasing aggressiveness, after selecting 500 examples.

The fact that the line goes down, then up again suggests that uncertainty sampling can be too aggressive. The 'somewhat aggressive' algorithm seems to be better than the really aggressive algorithm.

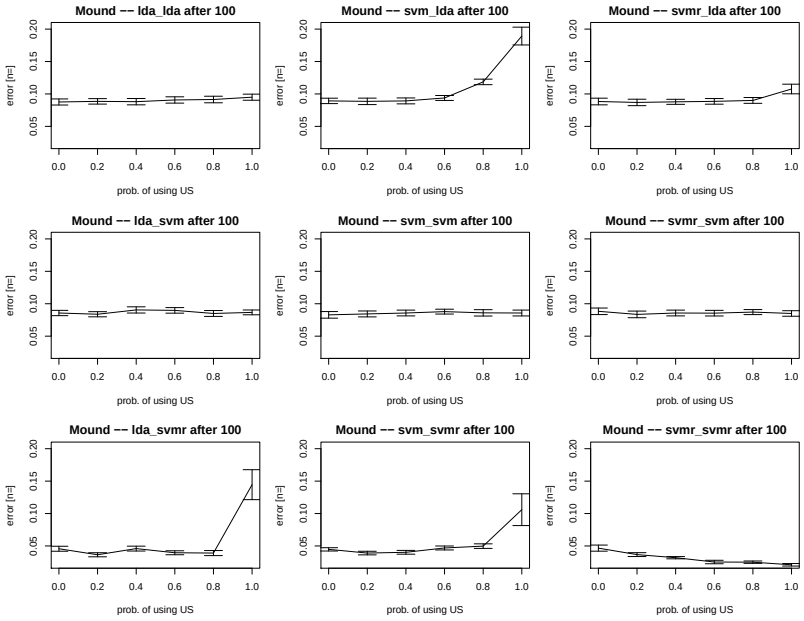


Figure 12: Errors of active learners against increasing aggressiveness, after selecting 100 examples.

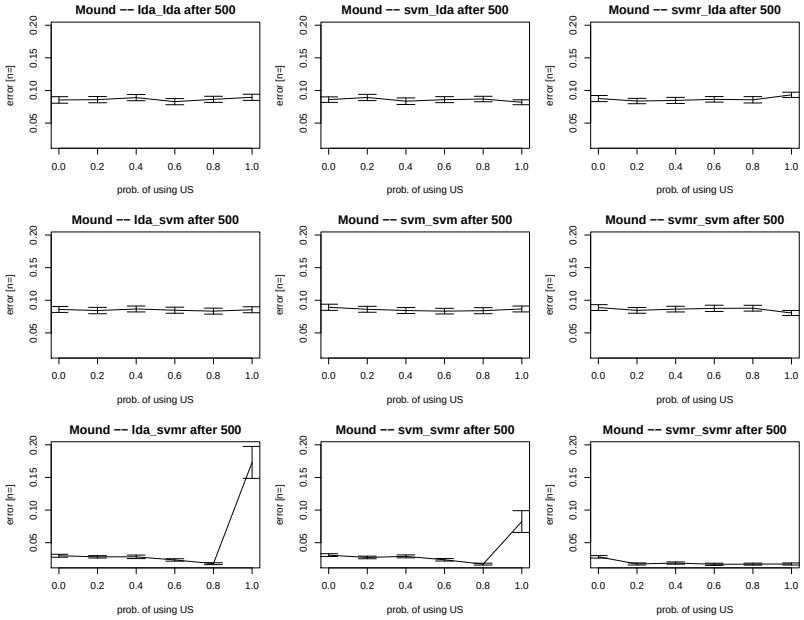


Figure 13: Errors of active learners against increasing aggressiveness, after selecting 500 examples.

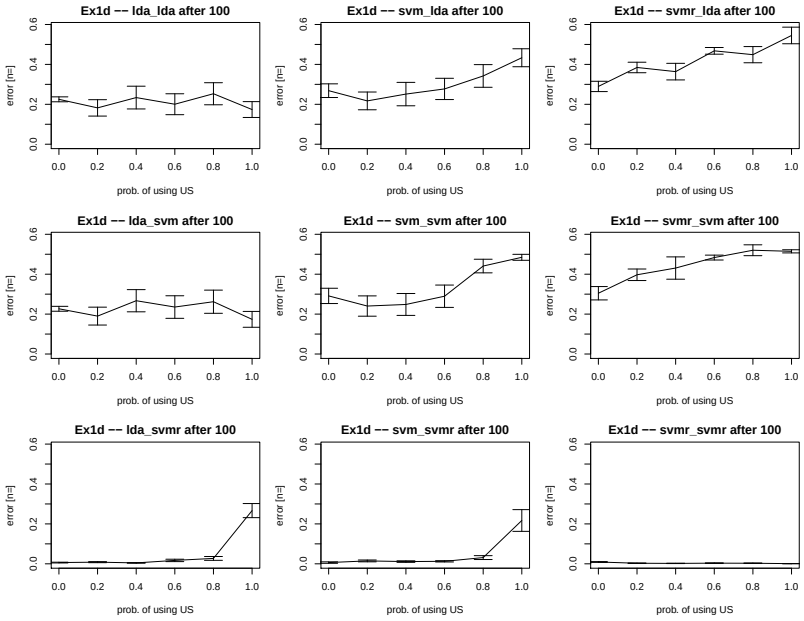


Figure 14: Errors of active learners against increasing aggressiveness, after selecting 100 examples.

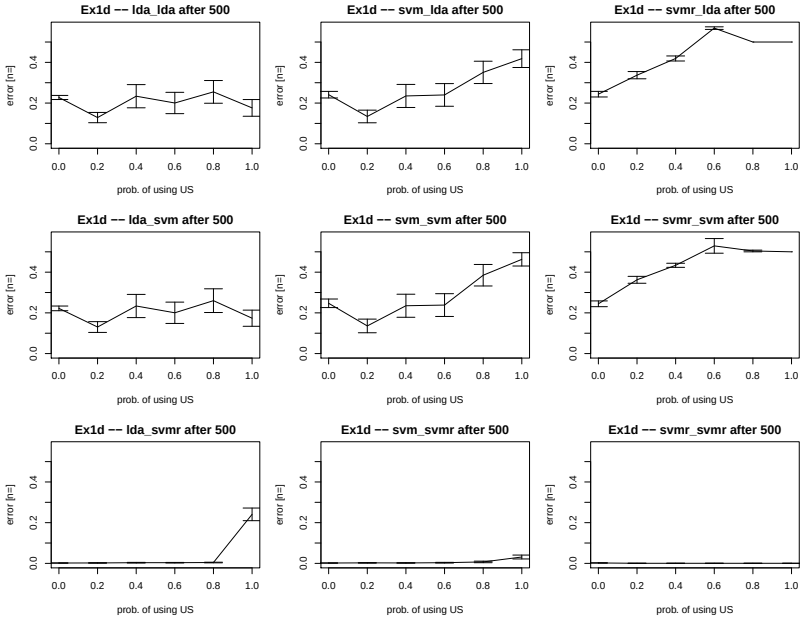


Figure 15: Errors of active learners against increasing aggressiveness, after selecting 500 examples.

7 Does AALWC solve the reusability problem?

The inventors of the AALWC algorithm suggest that it does, because it "creates an importance weighted sample that allows for unbiased risk estimation, even for hypotheses from a class different from the one employed by the active learner" (Beygelzimer et al., 2010).

1. Can this claim be empirically verified?
2. If it works, why? Is this only because of importance weighting, or are there other reasons (e.g., the fact that it is a consistent algorithm)?

1. Can this claim be empirically verified?

So far, the claim is just a sentence in the papers. It may be likely that it is true, because of the importance weighted risk estimation: if you apply it to another classifier the risk estimator is still unbiased. (I am not sure what it does to the deviation bounds.) It might be interesting to do Tomanek and Morik-style experiments with AALWC to see if this claim holds in practice.

2. If it works, why? Is this only because of importance weighting?

If AALWC creates a reusable sample only because of the importance weights, one would expect that other selection strategies can be made 'reusability-proof' by adding importance weights to the result. One of the advantages of AALWC is that it provides a natural way to find these weights. Since it is hard to estimate the weights otherwise, experiments could use an 'importance weighting oracle' that gives perfect weights.

But what are perfect weights? I wonder if density-based importance weights are sufficient. In AALWC the risk estimate is unbiased because the importance weights are the inverse of the selection probability. In active learning the selection probability is not the same as the density – for example, what is the selection probability in uncertainty sampling?

Even with perfect importance weights, I find it hard to believe that importance weights are enough to make the sample selection reusable. If that were true, it would imply that *any* selection strategy could be fixed, even uncertainty sampling. However, importance weighting cannot solve the missed cluster problem: importance weights do not help if areas are not sampled at all.

There may be another requirement apart from importance weights. Perhaps the algorithm should also be consistent?

Definitions of sample reuse and sample reusability from Tomanek (2010):

Definition 7.1 (Sample reuse). AL sample reuse describes a scenario where a sample S obtained by AL using learner T_1 during selection is exploited to induce a particular model type with learner T_2 with $T_2 \neq T_1$.

Definition 7.2 (Sample reusability). Given a random sample S_{RD}, and a sample S_{T_1} obtained with AL and a selector based on learner T_1, and a learner T_2 with $T_2 \neq T_1$. We say that S_{T_1} is reusable by learner T_2 if a model θ' learned by T_2 from this sample, i.e., $T_2\left(S_{T_1}\right)$, exhibits a better performance on a held-out test set $\mathcal{T}$ than a model θ'' induced by $T_2\left(S_{RD}\right)$, i.e., $\text{perf}\left(\theta', \mathcal{T}\right) > \text{perf}\left(\theta'', \mathcal{T}\right)$.

The AALWC algorithm (Beygelzimer et al., 2010) returns a hypothesis h_n that has "roughly the same error bound as a hypothesis returned by a standard passive learner":

Theorem 7.1. The following holds with probability at least $1 - \delta$. For any $n \geq 1$,

$$\text{err}\left(h_n\right) \leq \text{err}\left(h^*\right) + \sqrt{\frac{2C_0 \log n}{n-1}} + \frac{2C_0 \log n}{n-1}$$

8 *Solving s in AALWC*

A solution to equation 2 in Beygelzimer et al. (2010):

$$G_k = \left(\frac{c_1}{\sqrt{s}} - c_1 + 1\right) \cdot \sqrt{\frac{C_0 \log k}{k-1}} + \left(\frac{c_2}{s} - c_2 + 1\right) \cdot \frac{C_0 \log k}{k-1}$$

$$G_k = \left(\frac{c_1}{\sqrt{s}} - c_1 + 1\right) \cdot \sqrt{A} + \left(\frac{c_2}{s} - c_2 + 1\right) \cdot A \qquad \text{where } A = \frac{C_0 \log k}{k-1}$$

$$G_k = \frac{\sqrt{A}c_1}{\sqrt{s}} + (1 - c_1) \cdot \sqrt{A} + \frac{Ac_2}{s} + (1 - c_2) \cdot A$$

$$G_k - (1 - c_1) \cdot \sqrt{A} - (1 - c_2) \cdot A = \frac{\sqrt{A}c_1}{\sqrt{s}} + \frac{Ac_2}{s}$$

$$\gamma = \frac{\sqrt{A}c_1}{\sqrt{s}} + \frac{Ac_2}{s} \qquad \text{where } \gamma = G_k - (1 - c_1) \cdot \sqrt{A} - (1 - c_2) \cdot A$$

Solve for s with Wolfram Alpha

$$s = \frac{\alpha^2 \pm \sqrt{\alpha^4 + 4\alpha^2\beta\gamma} + 2\beta\gamma}{2\gamma^2} \qquad \text{where } \alpha = \sqrt{A}c_1 \text{ and } \beta = Ac_2$$

The algorithm further requires that $s \in (0, 1)$.

9 *Experiments with reusability in AALWC*

I did (and am still doing) reusability experiments with the AALWC algorithm (Beygelzimer et al., 2010), comparing the errors of different selectors and consumers for different C_0 parameters. The plots on the following pages show the results I found so far.

There are a few problems I encountered:

1. **How to compare different runs of the algorithm?**

   The parameter C_0 influences the number of examples that the algorithm selects. However, this number of selected examples is not fixed: it differs per selector and per run. What is a 'fair' comparison? Can you compare the error of the `lda` selector, which selects 100 to 200 examples at $C_0 = 8$, with that of the `svmr` selector, which selects up to 1000 examples with the same C_0?

   A similar problem: what is the 'random sampling' equivalent to compare with? In the current experiments, I compare the error of an active selection with the error on a random selection with the same number of labeled examples. But since active learner has seen *all* unlabeled examples, and only labeled a few, perhaps the active error should be compared with the error of the classifier trained on the full set of labeled examples.

   The AALWC paper provides deviation bounds of the performance relative to a passive learner that has seen the same number of examples as the active learner (so the passive learner would get as many labeled examples as the active learner would get unlabeled examples). In the experiments I have done so far the active selection performs worse than random selection.

2. **Which error should be minimised by the ERM oracle?**

   In my experiments I found that quite often the 'alternative hypothesis' had a lower importance-weighted error on the labeled training set than the 'current hypothesis', that is, $\operatorname{err}\left(h'_k, S_k\right) < \operatorname{err}\left(h_k, S_k\right)$. This is not what the algorithm expects, since:

   $$h_k = \arg\min_{h \in \mathcal{H}} \operatorname{err}\left(h, S_k\right) \qquad h'_k = \arg\min_{h \in \mathcal{H} \wedge h(x_k) \neq h_k(x_k)} \operatorname{err}\left(h, S_k\right)$$

   The problem is probably that the classifiers do not minimise the error on the training set, but try to minimise the expected error on the true distribution instead. For example, SVM has its regularisation term. To get a good definition of the errors in the algorithm, I should probably ask the classifiers to find a minimum on the training set: for example, by setting the cost of misclassifications of the SVM algorithm to something very high.

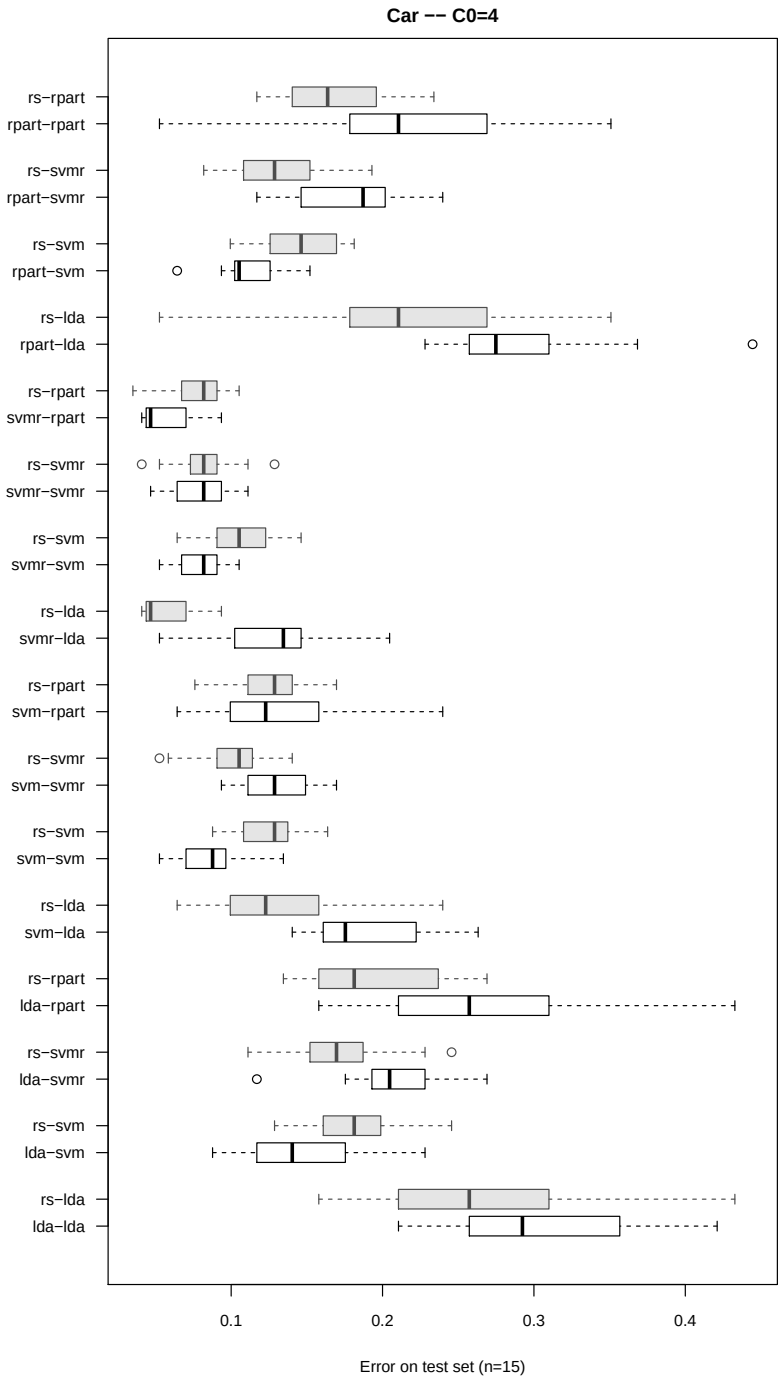


Figure 16: Reusability experiments with the UCI `car` dataset.
The graph shows the errors on the test set of different selector/consumer pairs; `lda-svm` indicates that LDA was used to select the examples and SVM was used to train the final classifier.
Classifiers: linear discriminant analysis (`lda`), support vector machines with a linear kernel (`svm`) and with a radial-basis function (`svmr`), decision trees (`rpart`). For comparison, the `rs-*` results show the errors of the consumer trained with randomly selected samples (using the same number of examples as the active selection). With reusability, the error of the active selection is expected to be lower than that of random selection.
Note that the number of selected examples is different in each experiment: there is a large difference between different selectors (for the same C_0 parameter) and a smaller difference between different runs of one selector. Figure 21 shows the distribution of these numbers.
AALWC parameters: $C_0 = 4$, $c_1 = c_2 = 1$.

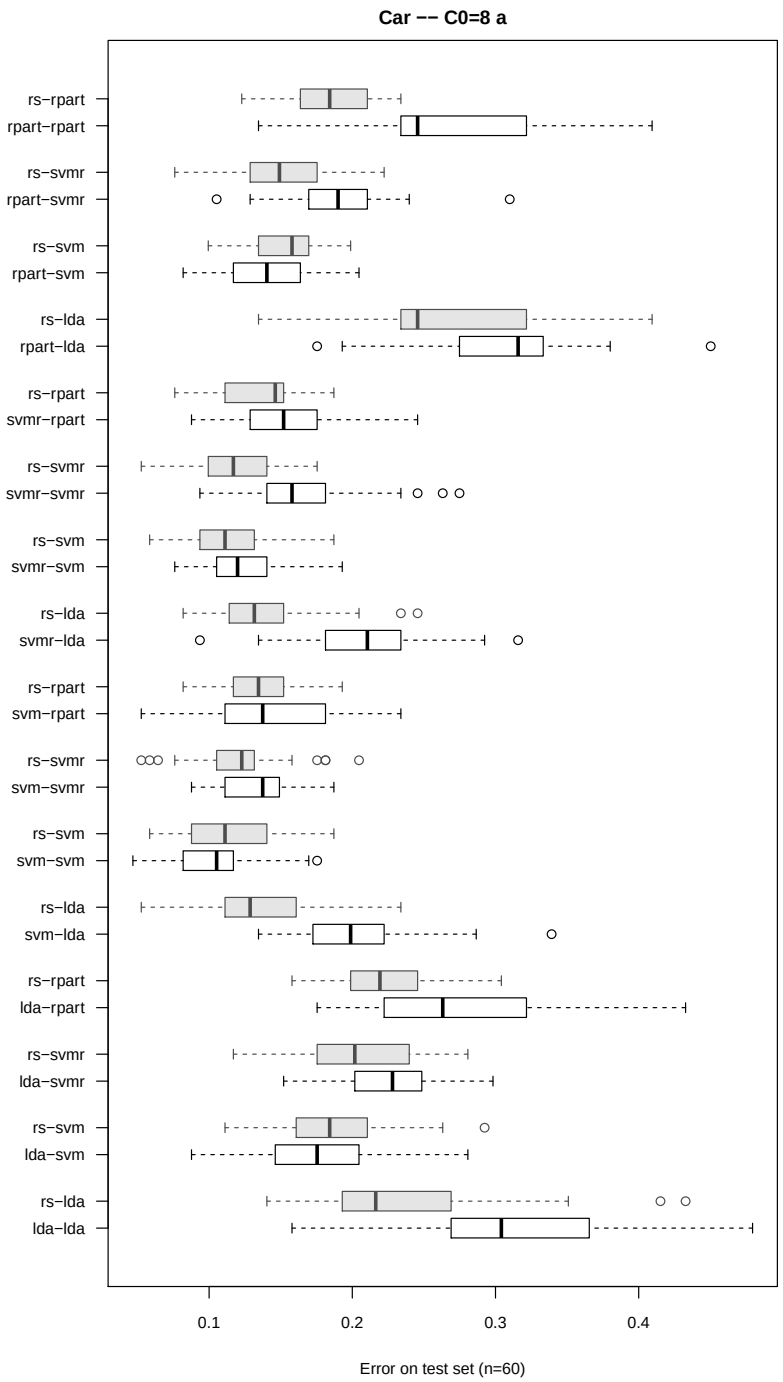


Figure 17: Reusability experiments with the UCI `car` dataset. Similar to figure 16, but with $C_0 = 8$.
Note: something is wrong with these experiments. The number of selected examples is much lower than expected. See figure 21 for more information. Figure 18 shows experiments with $C_0 = 8$ that seem more plausible.

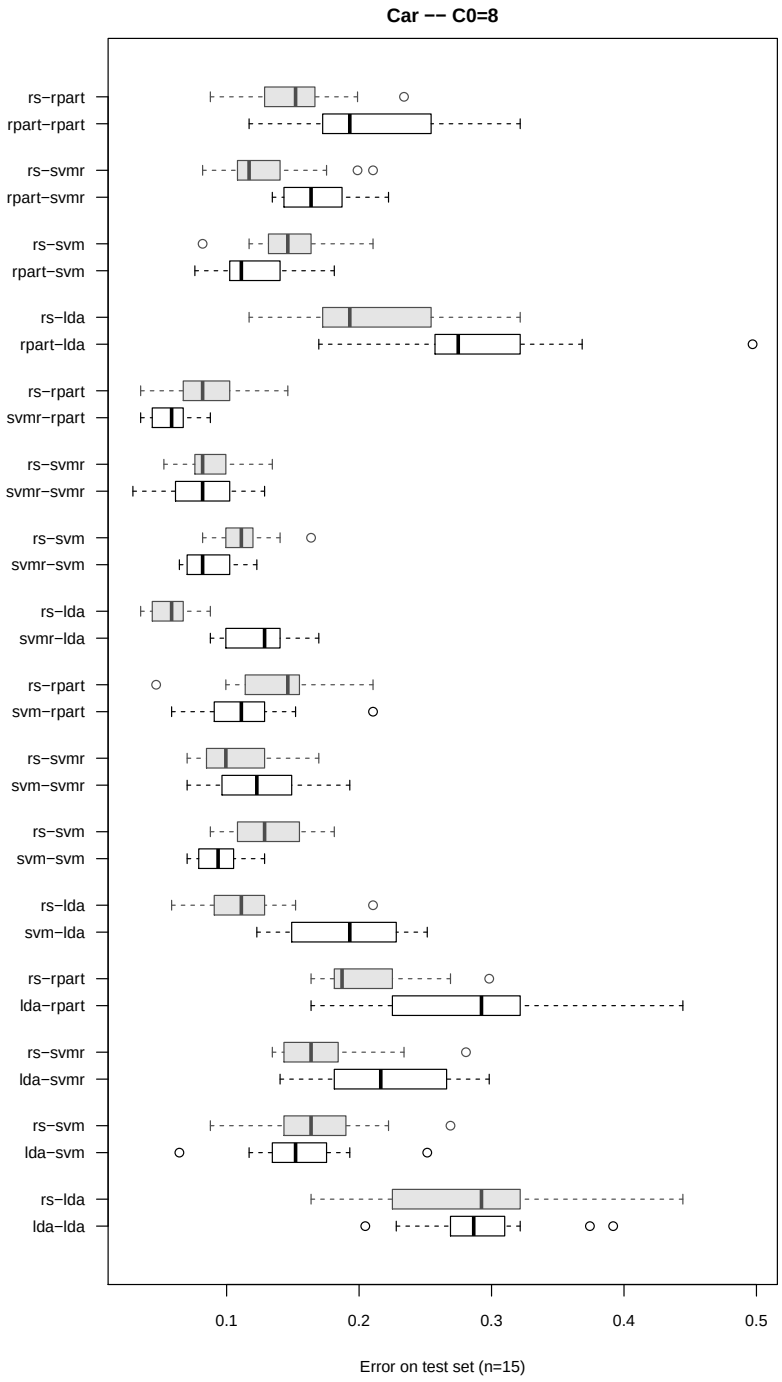


Figure 18: Reusability experiments with the UCI `car` dataset. Similar to figure 16, but with $C_0 = 8$.

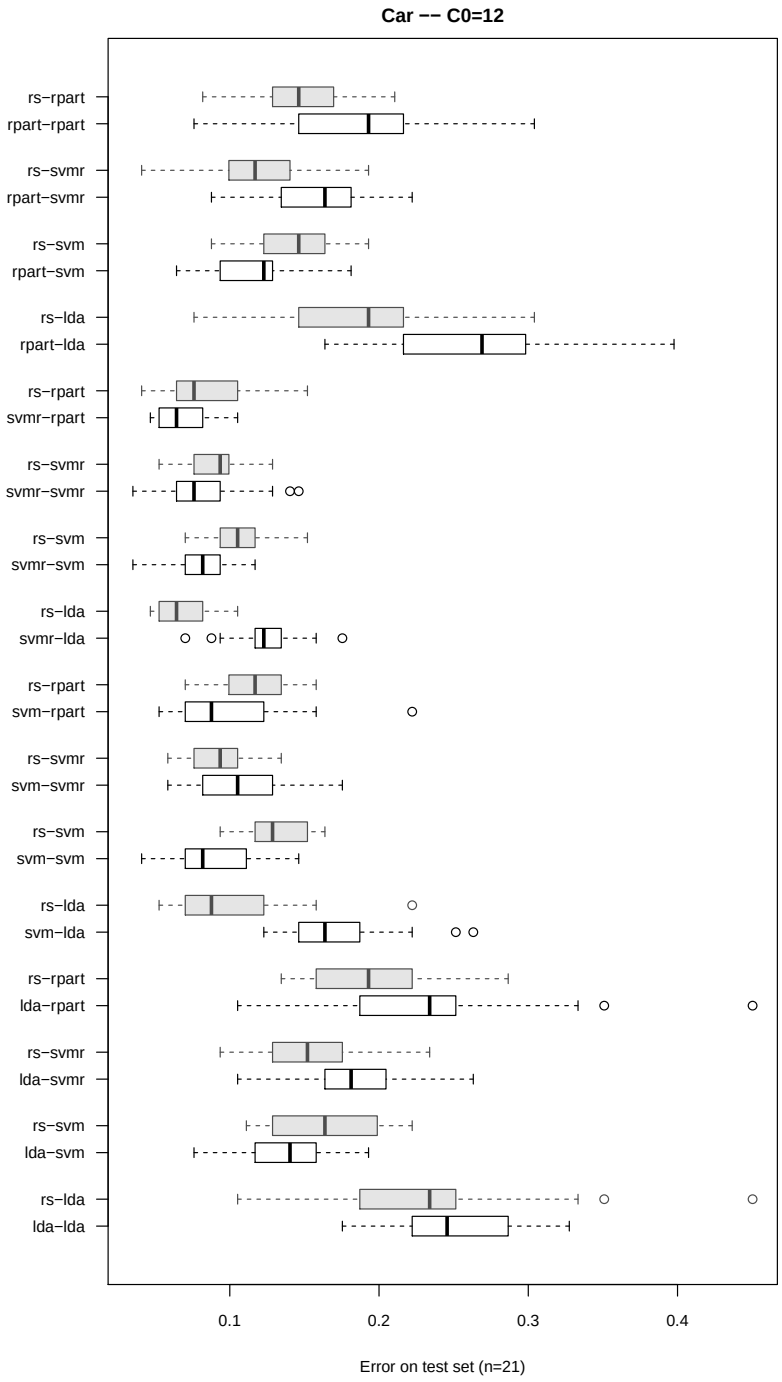


Figure 19: Reusability experiments with the UCI `car` dataset. Similar to figure 16, but with $C_0 = 12$.

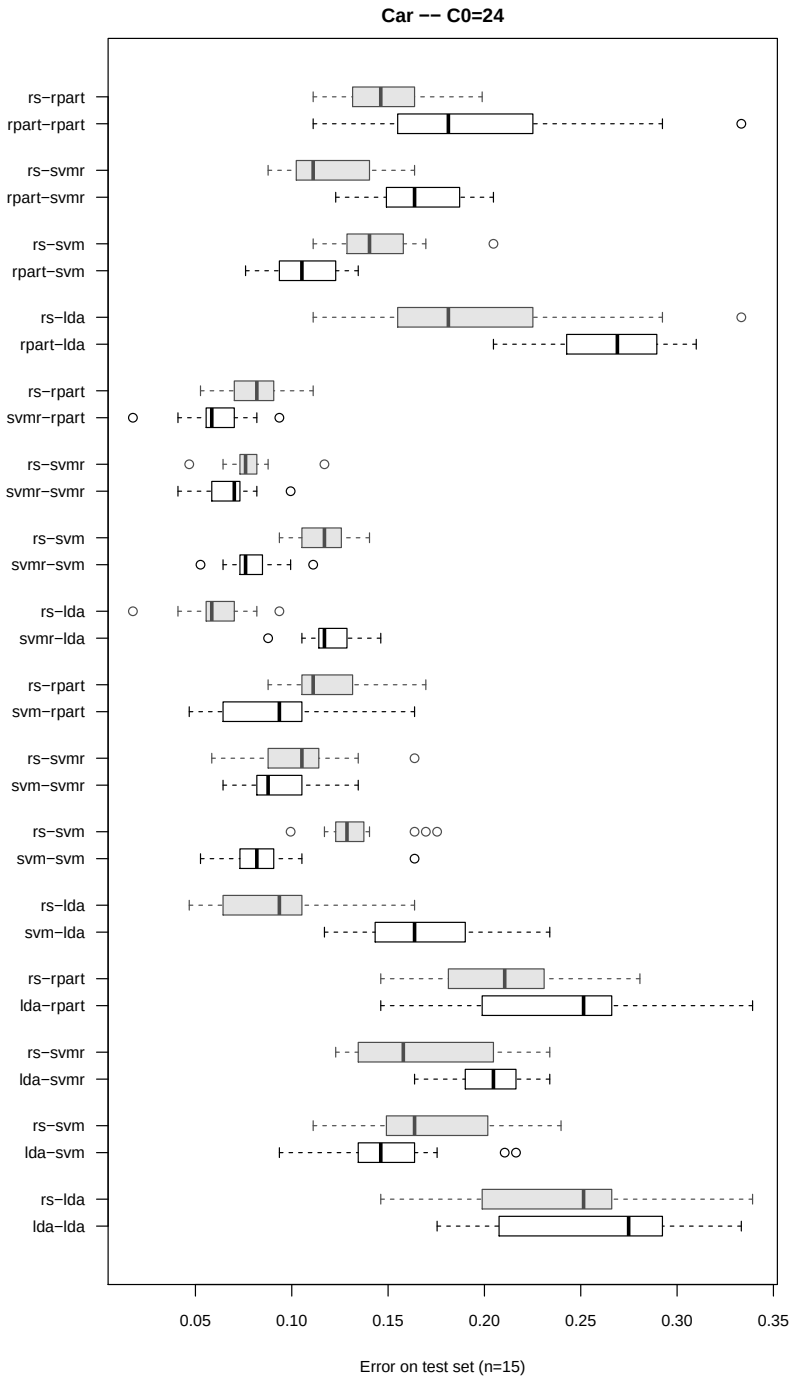


Figure 20: Reusability experiments with the UCI `car` dataset. Similar to figure 16, but with $C_0 = 24$.

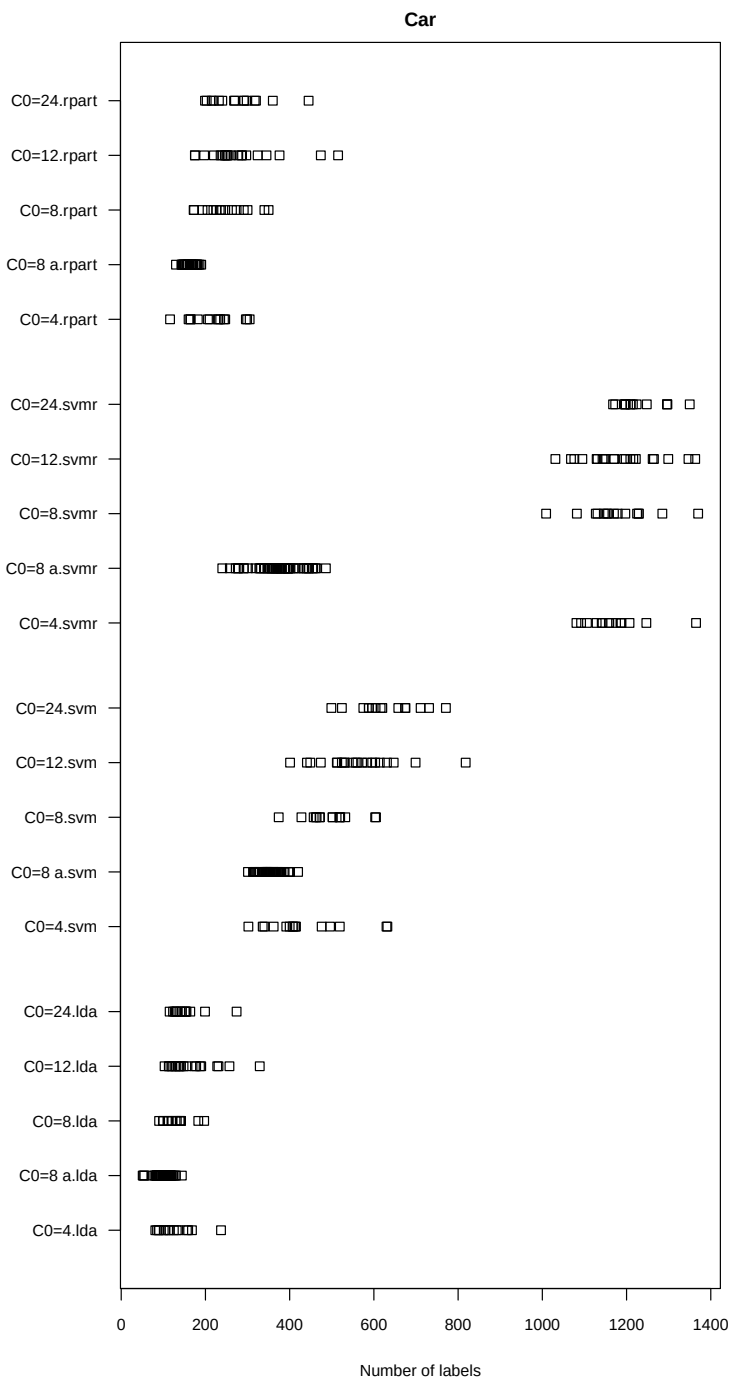


Figure 21: Reusability experiments with the UCI `car` dataset.
This plot shows the number of examples that were selected by each selector.
The C_0 parameter of the AALWC algorithm determines the number of examples: the higher C_0, the more likely the algorithm is to select an example.
My first round of experiments ($C_0 = 8$, marked with `C0=8 a` in this plot) is different from the other runs. As can be seen in the plot, the numbers are clearly different. I must have changed something in the code; a new run with the same parameter (`C0=8`) shows results that are much more likely.

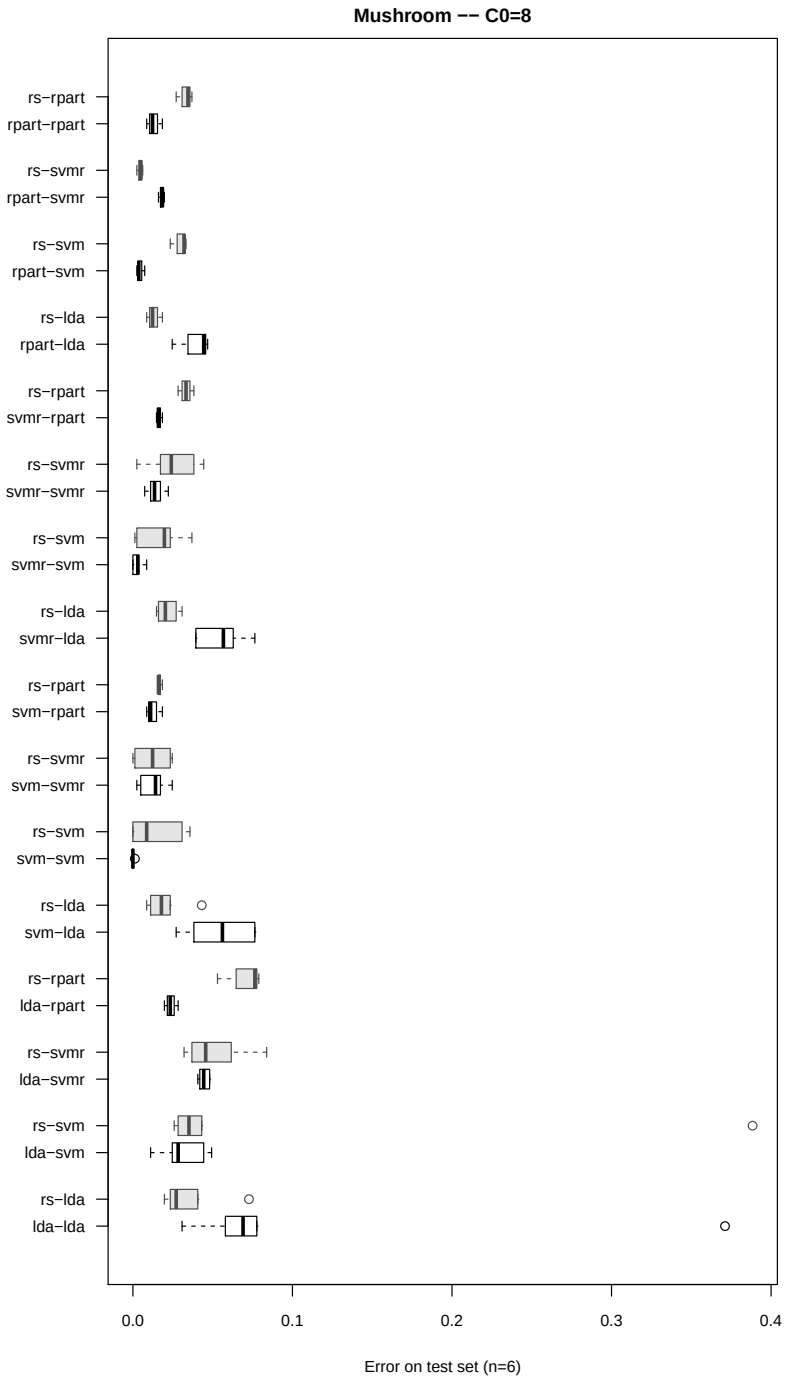


Figure 22: Reusability experiments with the UCI `mushroom` dataset. $C_0 = 8$ The larger size of `mushroom` means that experiments with this dataset take much longer. However, these initial results suggest that the reusability of some pairs is different from that on `car`.

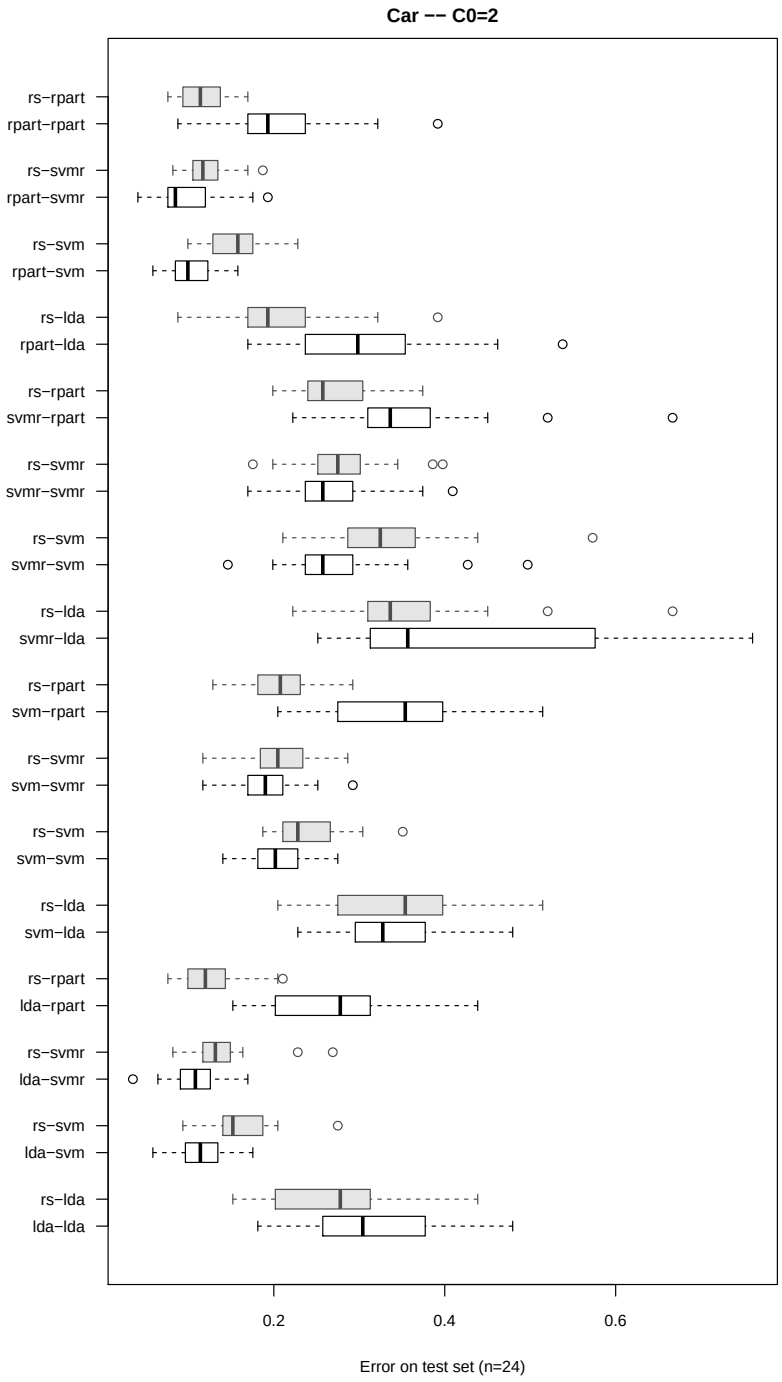


Figure 23: Reusability experiments with the UCI `car` dataset.
Similar to the previous experiments, but with the `cost` parameter of the `svm` and `svmr` classifiers set to 1000, to have the SVM algorithm minimise the error on the training set.

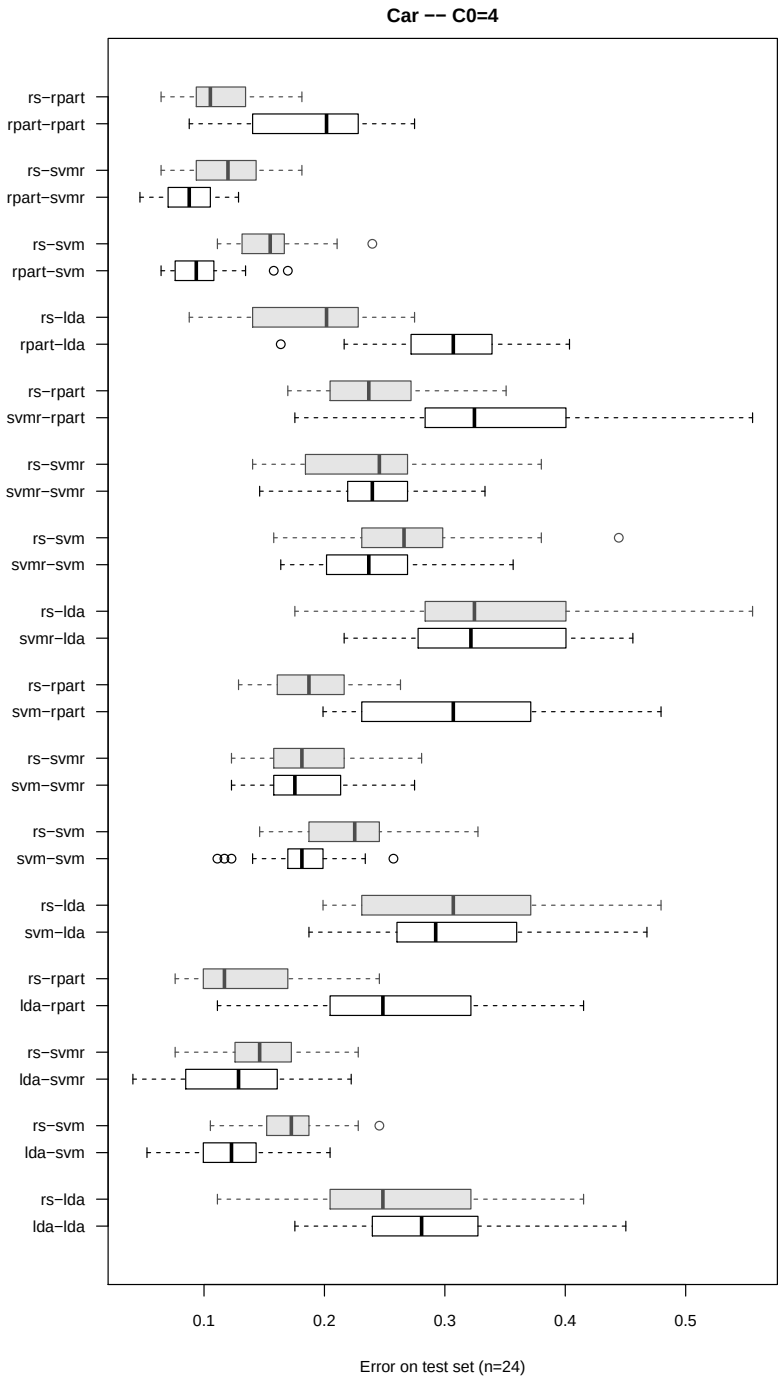


Figure 24: Reusability experiments with the UCI `car` dataset.
Similar to the previous experiments, but with the `cost` parameter of the `svm` and `svmr` classifiers set to 1000, to have the SVM algorithm minimise the error on the training set.

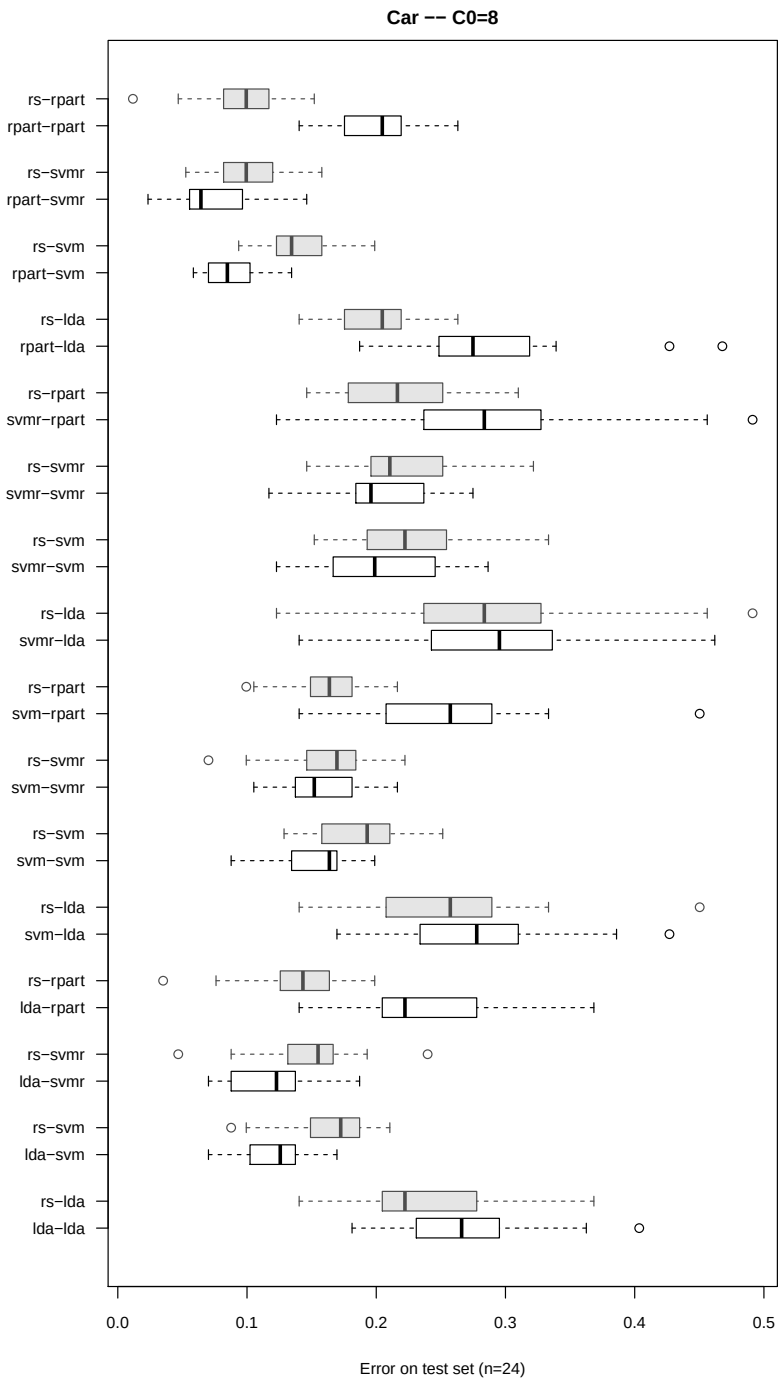


Figure 25: Reusability experiments with the UCI `car` dataset.
Similar to the previous experiments, but with the `cost` parameter of the `svm` and `svmr` classifiers set to 1000, to have the SVM algorithm minimise the error on the training set.

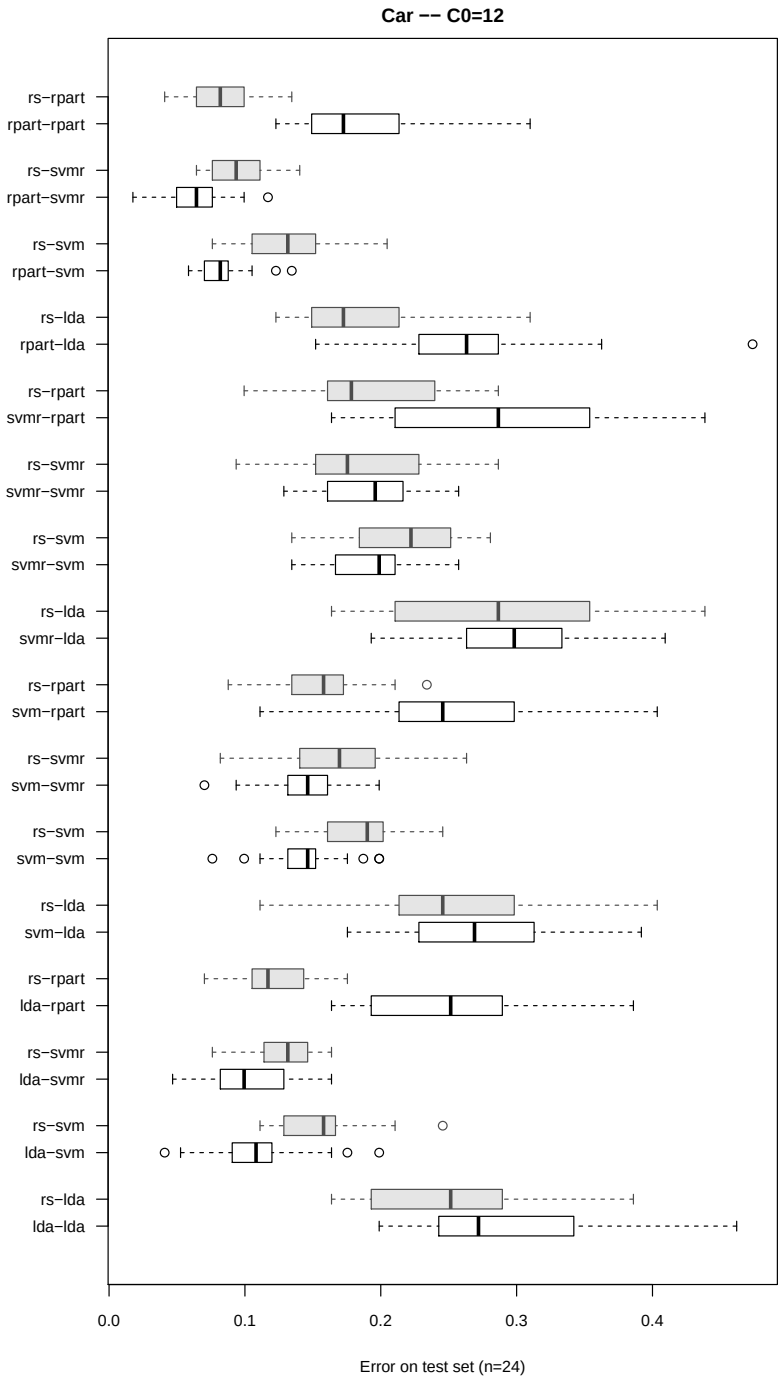


Figure 26: Reusability experiments with the UCI `car` dataset.
Similar to the previous experiments, but with the `cost` parameter of the `svm` and `svmr` classifiers set to 1000, to have the SVM algorithm minimise the error on the training set.

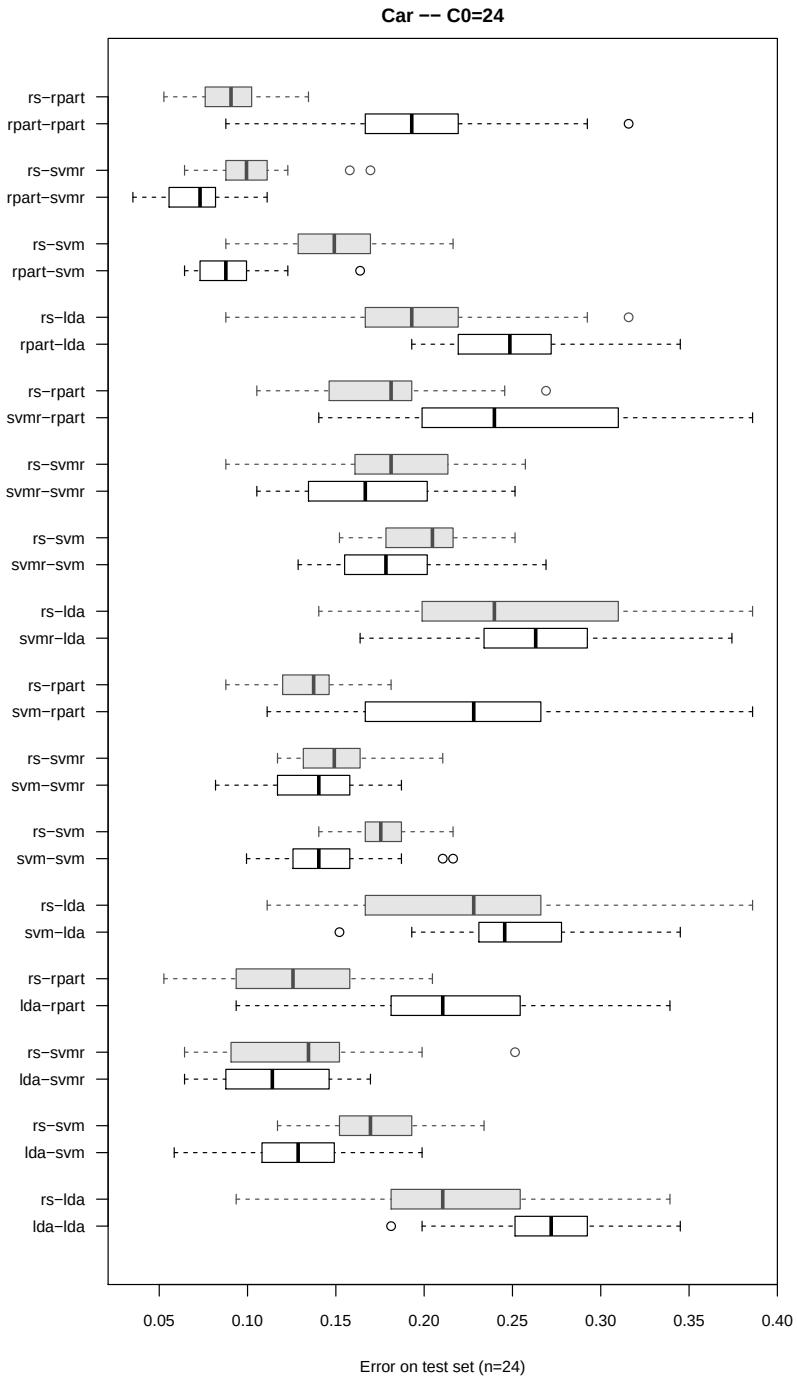


Figure 27: Reusability experiments with the UCI `car` dataset.
Similar to the previous experiments, but with the `cost` parameter of the `svm` and `svmr` classifiers set to 1000, to have the SVM algorithm minimise the error on the training set.

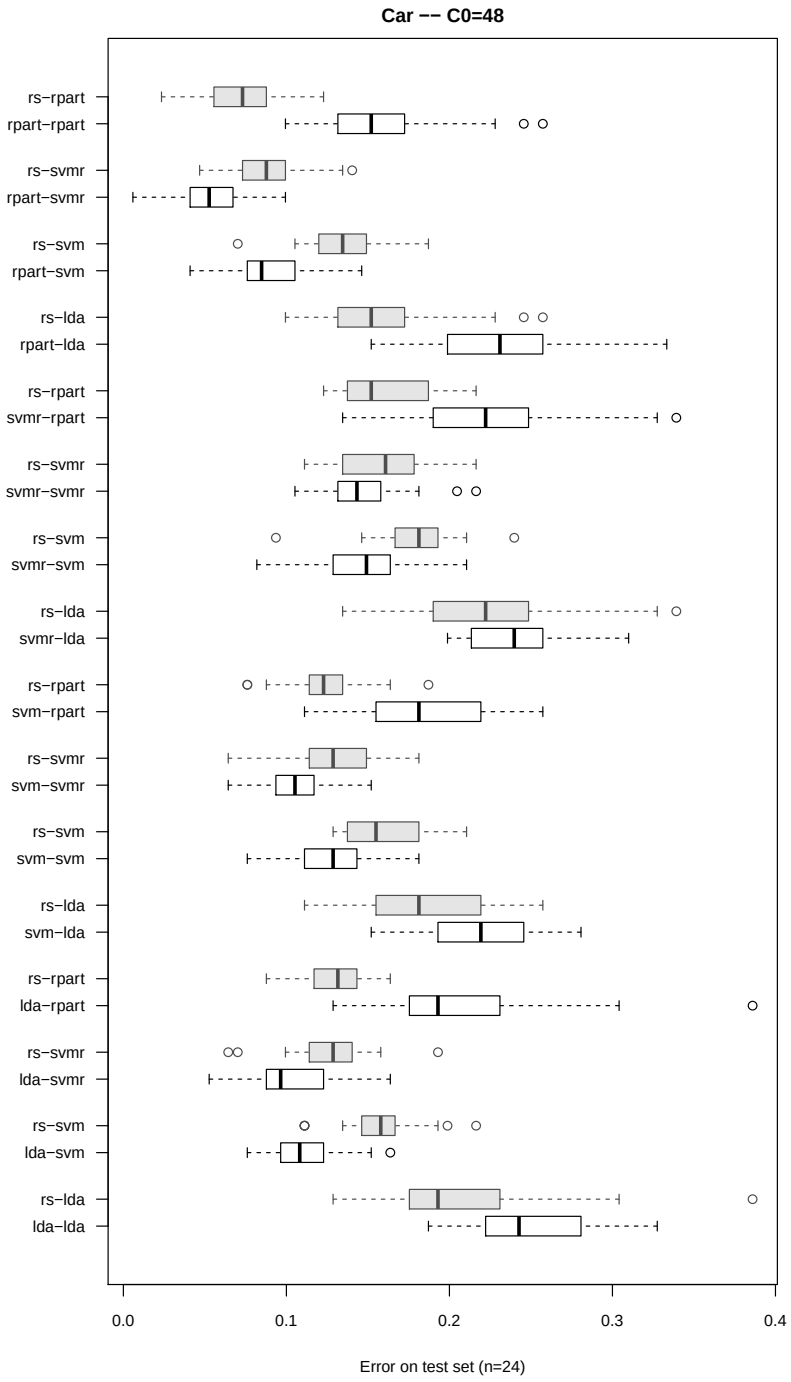


Figure 28: Reusability experiments with the UCI `car` dataset.
Similar to the previous experiments, but with the `cost` parameter of the `svm` and `svmr` classifiers set to 1000, to have the SVM algorithm minimise the error on the training set.

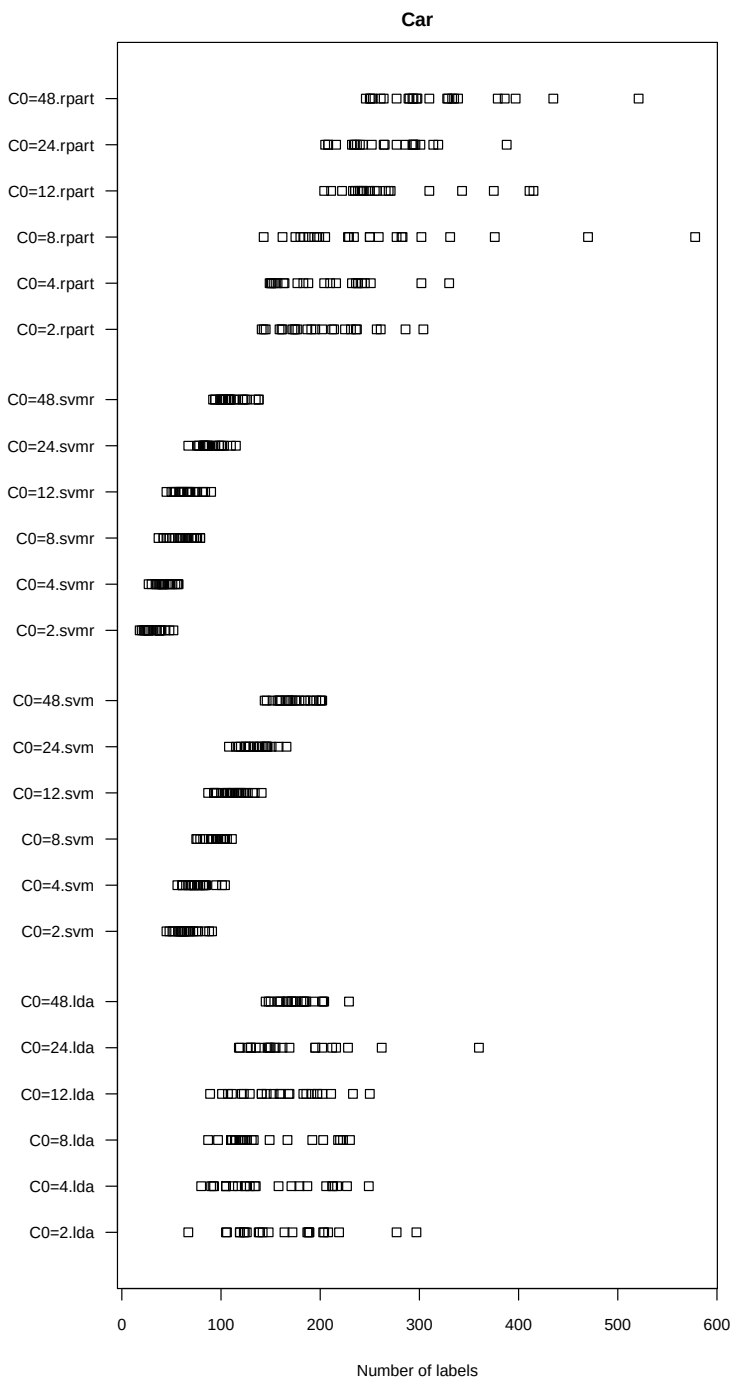


Figure 29: Reusability experiments with the UCI `car` dataset.
The number of selected examples for each setting of the C_0 parameter.
Similar to the previous experiments, but with the `cost` parameter of the `svm` and `svmr` classifiers set to 1000, to have the SVM algorithm minimise the error on the training set.

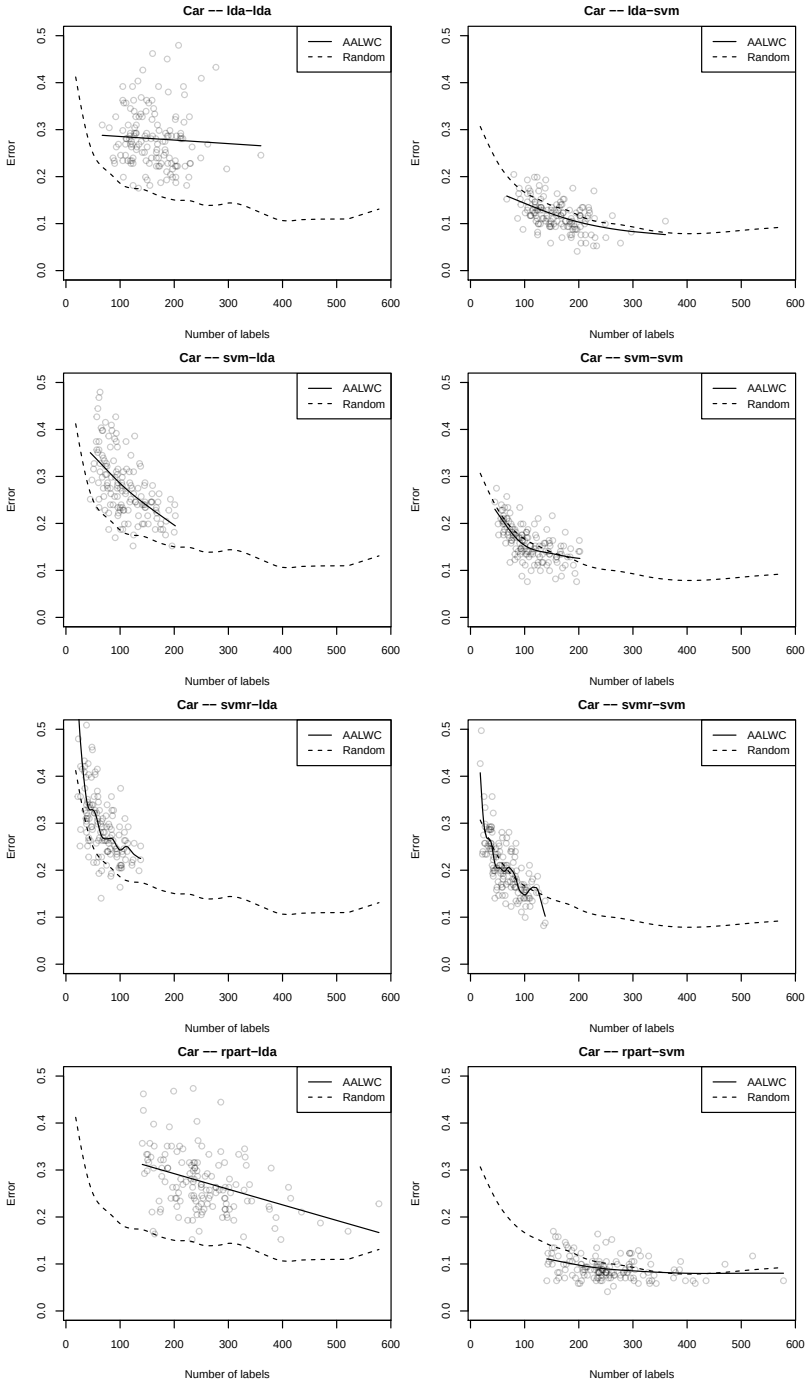


Figure 30: Reusability experiments with the UCI `car` dataset.
The (smoothed) error of the AALWC selection for different selector-consumer pairings, compared with random sampling with the same number of labeled examples. Active learning makes sense if the active learning error is lower than the error achieved with random sampling.

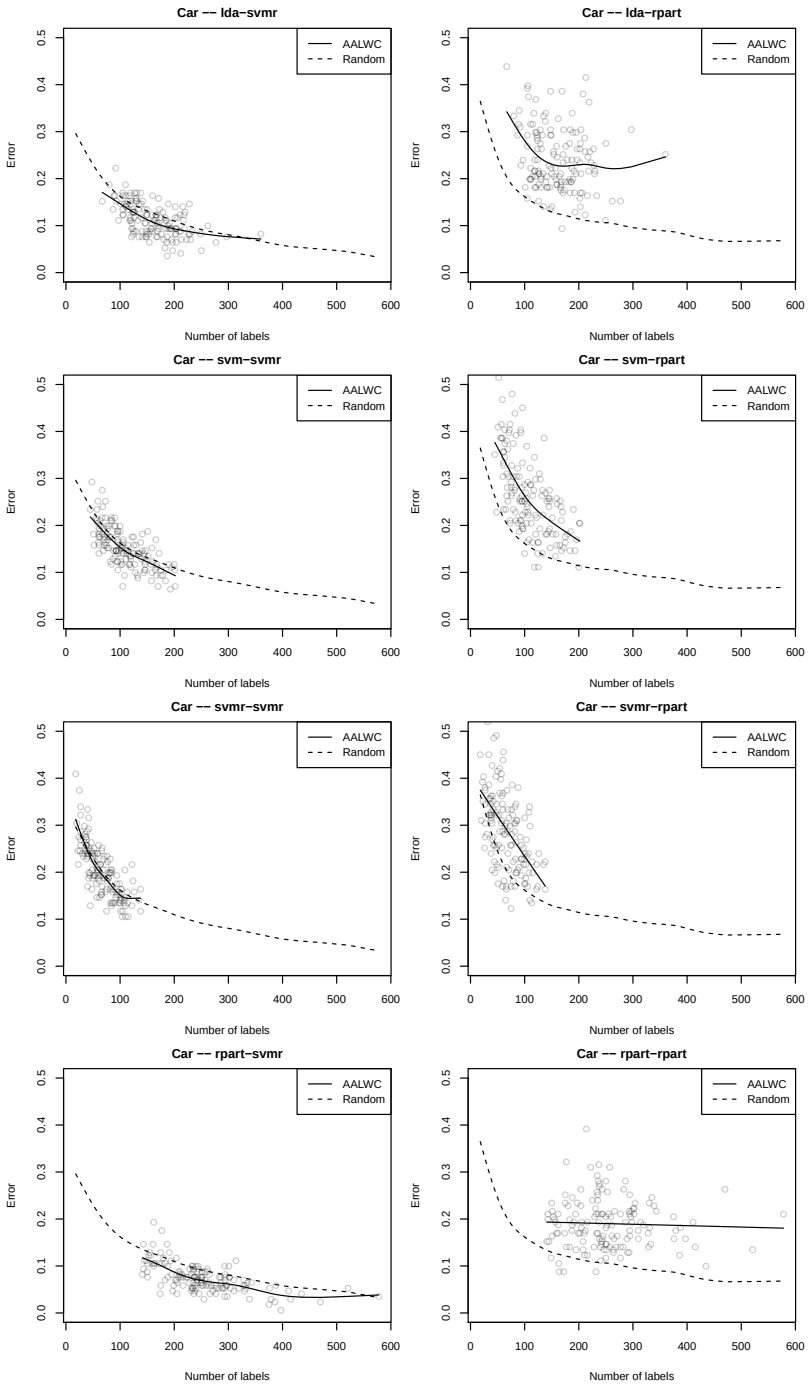


Figure 31: Reusability experiments with the UCI `car` dataset.
The (smoothed) error of the AALWC selection for different selector-consumer pairings, compared with random sampling with the same number of labeled examples. Active learning makes sense if the active learning error is lower than the error achieved with random sampling.

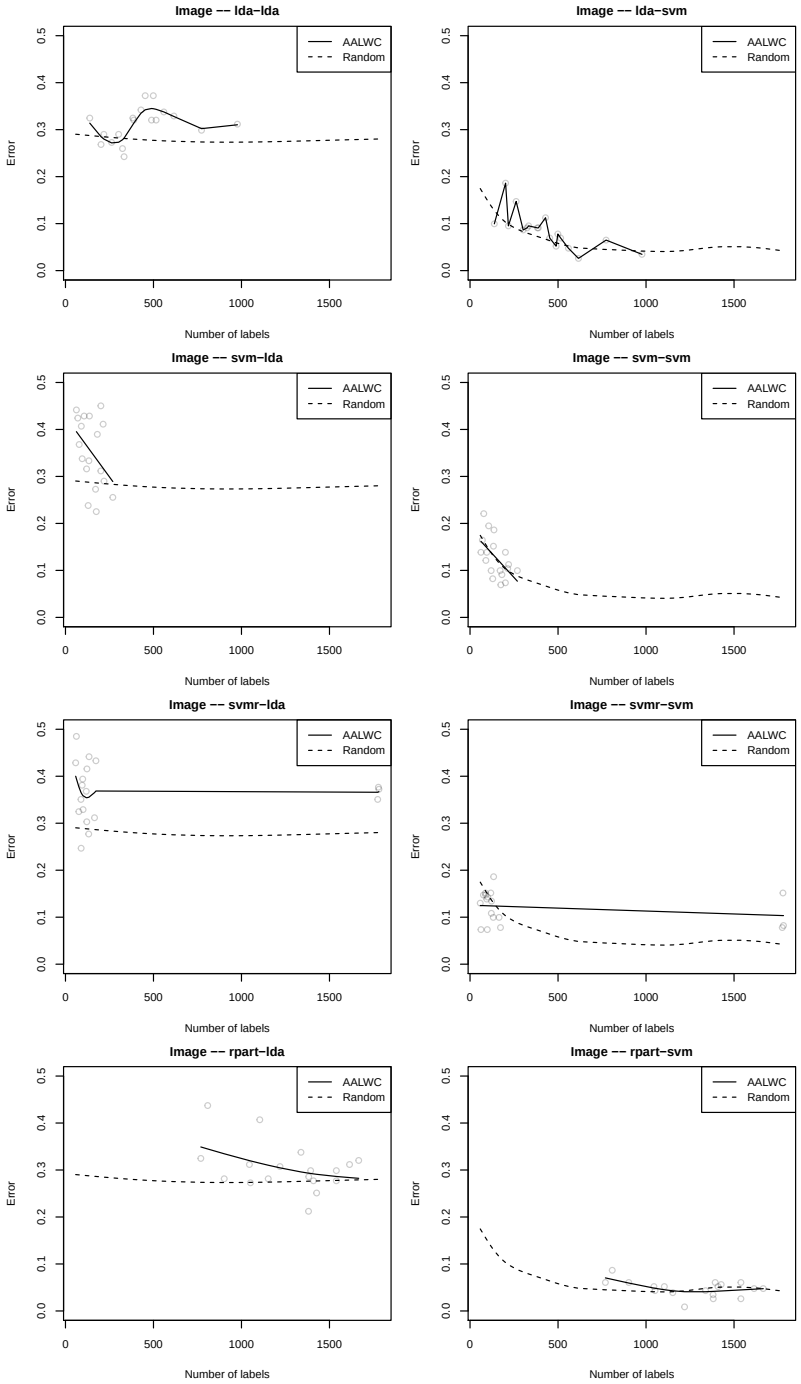


Figure 32: Reusability experiments with the UCI `image` dataset. The (smoothed) error of the AALWC selection for different selector-consumer pairings, compared with random sampling with the same number of labeled examples. (Preliminary results with only a few experiments.)

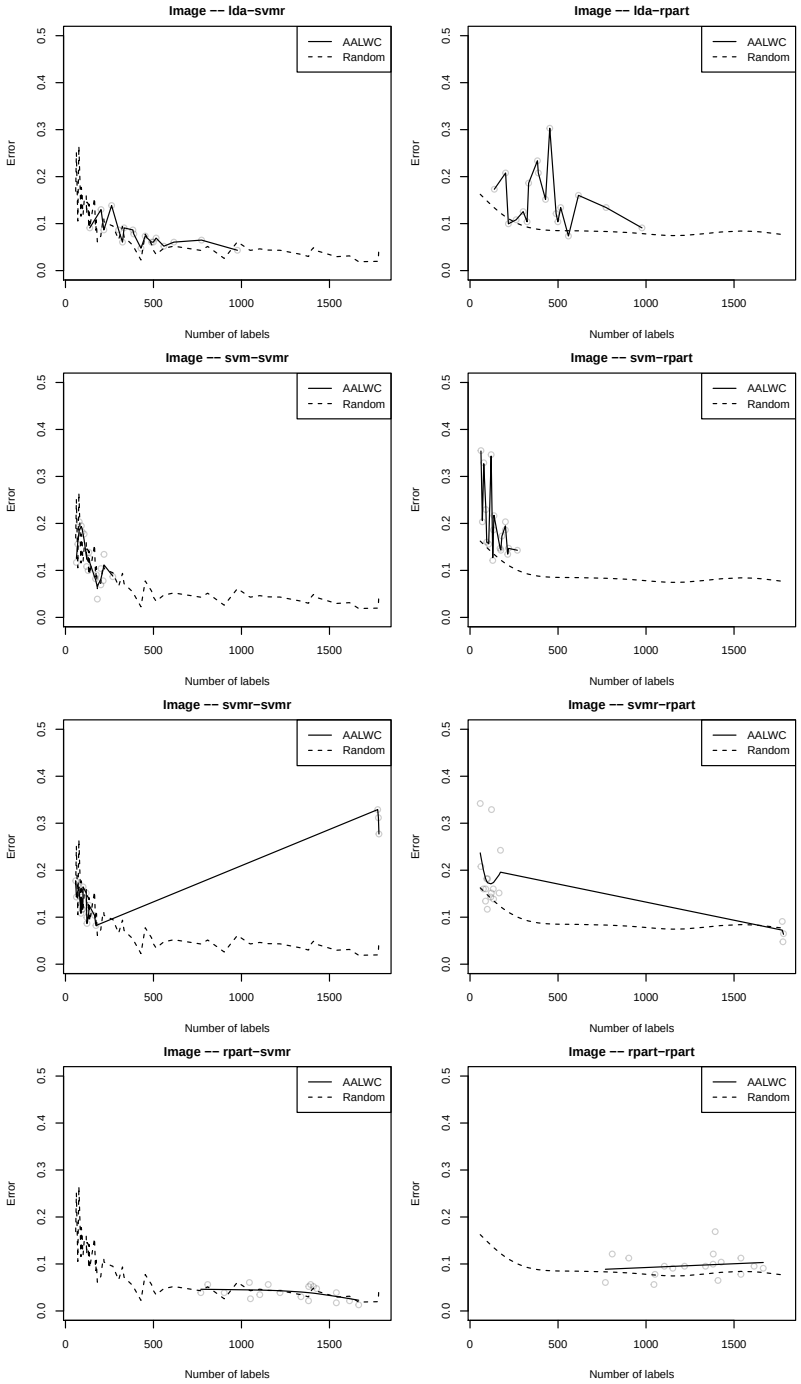


Figure 33: Reusability experiments with the UCI `image` dataset.
The (smoothed) error of the AALWC selection for different selector-consumer pairings, compared with random sampling with the same number of labeled examples. (Preliminary results with only a few experiments.)

10 Experiments with reusability in Vowpal Wabbit

I have done experiments with the Vowpal Wabbit, a fast online learning system. VW uses a sparse gradient descent algorithm to learn a linear model based on a squared loss function. An interesting aspect of VW is that it implements an importance-weighted active learning algorithm based on the paper by Beygelzimer et al. (2010). The implementation is not exactly the same: it uses an approximation of the error.

The source code of the Vowpal Wabbit and details about its implementation are available at `https://github.com/JohnLangford/vowpal_wabbit/`.

First experiments

First, I ran the Vowpal Wabbit on the UCI `car` dataset to see if the basic properties work. Figure 34 shows that, indeed, the active learner improves the learning rate of the classifier.

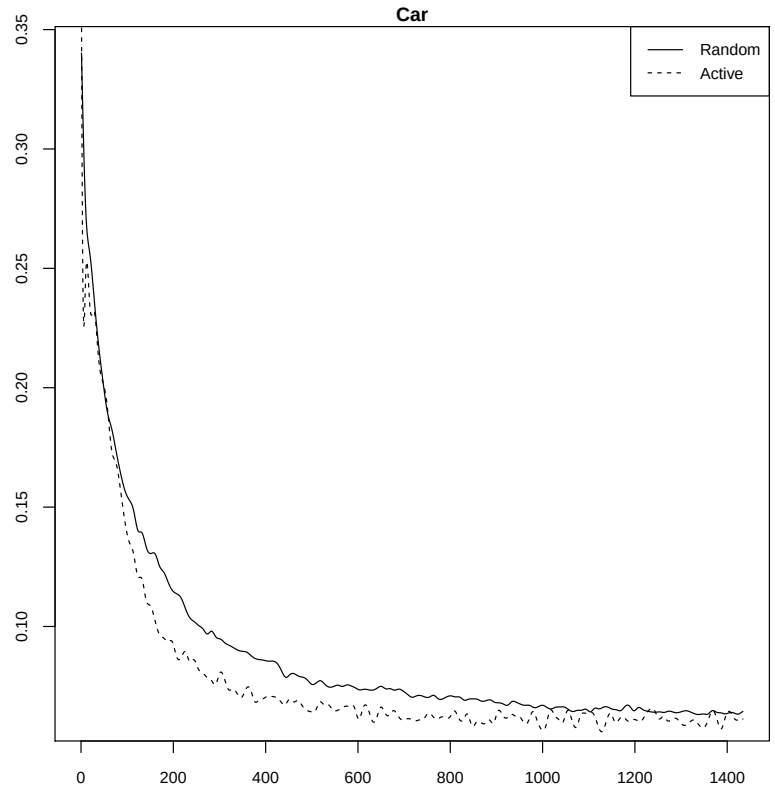


Figure 34: The result of several runs of the Vowpal Wabbit on the UCI `car` dataset.

To investigate the reusability aspects of the VW algorithm, I needed more than one classifier. The VW offers a quadratic option that can be used to model interactions between two features. I used this to compare the linear classifier with a quadratic classifier. Unfortunately, on my dataset the classifiers seemed to be too similar: there was little difference between the sample selections and the performances of both classifiers.

VW extensions

For further experiments, I decided to extend the VW source code.

First, I added a set of 'kernel functions' that map the incoming features

to a different feature space. In this way the linear gradient descent learner could still represent different types of classifiers. My kernel functions turn the learning problem into a distance-based learning problem: there is a feature for every example, representing the difference between that other example and the current example. This complicates the active learning problem: new features become available in each iteration, so the complete model has to be retrained in each step. I chose a 'Gaussian' kernel function $e^{-\|x-y\|^2}$ and a 'sigmoid' kernel function $\tanh(\gamma \cdot \langle x, y \rangle - c)$.

Next, I added an uncertainty sampling method to the Vowpal Wabbit and wrote a number of functions to run my experiments.

Results

The following figures show results of my experiments. I did experiments with the different kernels and the different selection strategies on the `car` dataset. The results seem to support my expectation that IWAL-sampling gives better reusability than uncertainty sampling.

In a second series of experiments, I used a letter recognition dataset. Instead of using different kernel functions, I looked at classifiers with different features. The letter dataset has features that work in the x and y directions, so I decided to see what happens if a selector with only x features selects examples for a consumer with y features. Again the results, shown in figure 38, suggest that IWAL is good for reusability.

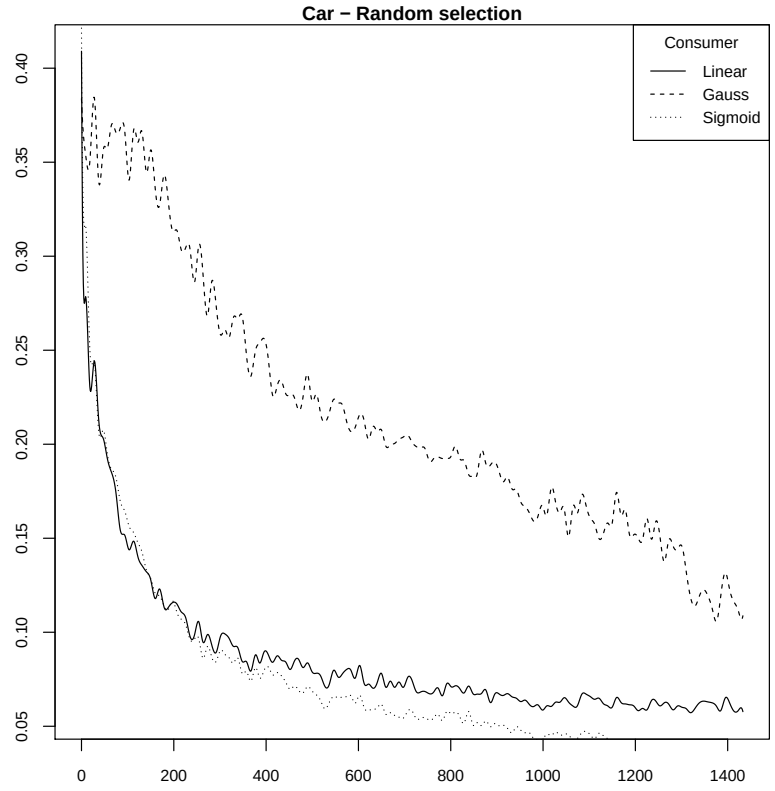


Figure 35: The UCI `car` dataset with different 'kernels', with random sampling. The Gaussian function is not very good, but the sigmoid function works quite well.

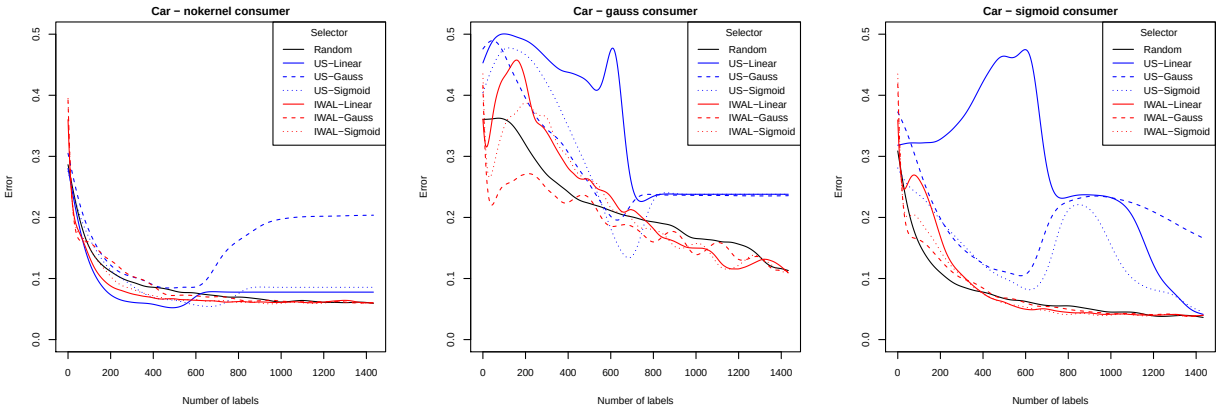


Figure 36: The UCI `car` dataset with different 'kernels' and different sampling methods. The lines suggest that the reusability of the IWAL-sampled data is better than that of uncertainty sampling.

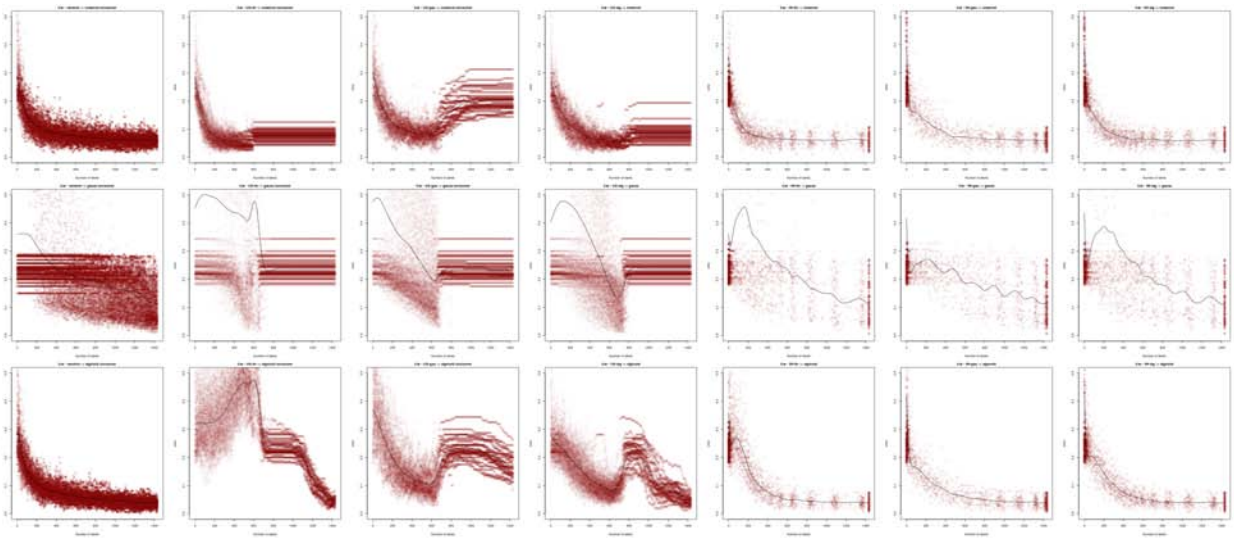
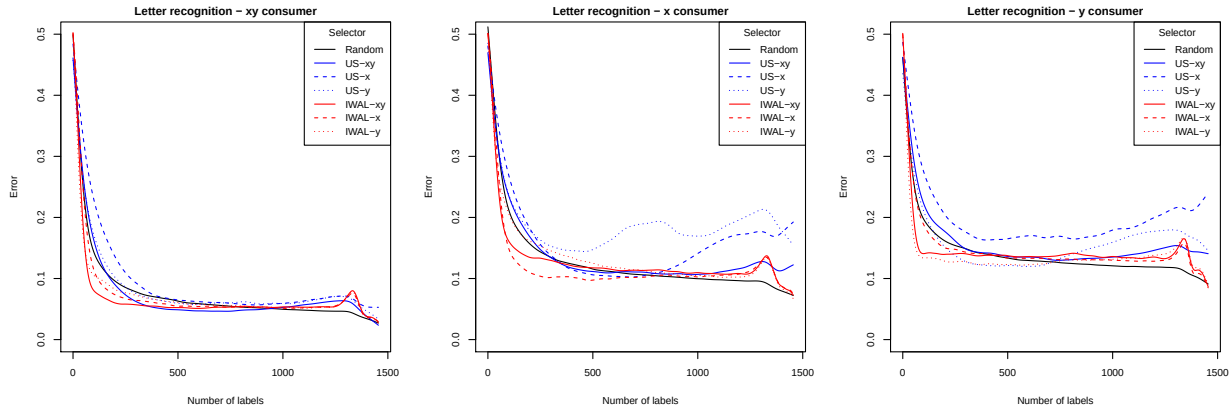


Figure 37: The range of the errors on the UCI `car` dataset with different 'kernels' and different sampling methods. Columns: random, US-linear, US-gauss, US-sigmoid, IWAL-linear, IWAL-gauss, IWAL-sigmoid selectors. Rows: linear, gauss, sigmoid consumers.



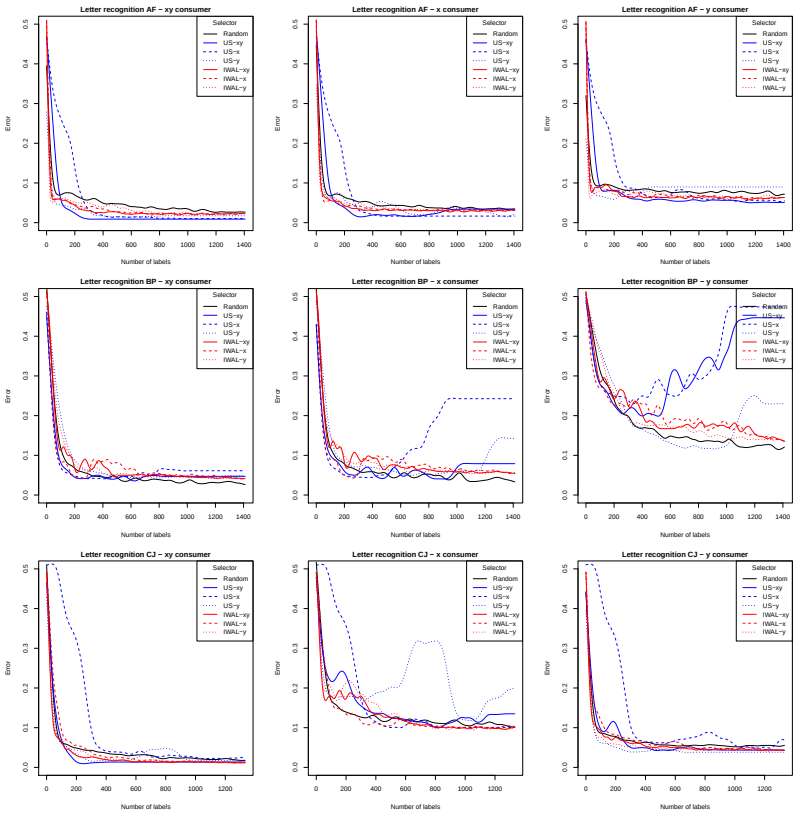


Figure 38: Reusability experiments on the letter recognition dataset. Classifiers based on only x or y features or both x and y. This graph shows the average of many such graphs for different letter combinations. The results vary per combination, but the reusability with uncertainty sampling is generally poor, while the IWAL results are often close to or better than random sampling.
To the left: a few of the graphs of individual letter combinations.

11 A question

How reusable are the samples selected by the Vowpal Wabbit IWAL algorithm? For large and small sample sizes.

The reusability claims of the IWAL algorithms come from their importance weighting. If the importance weights are such that the original distribution can be reconstructed, any learning algorithm should be able to learn from the active learning dataset.

My current impression is that good importance weights are a requirement for reusable active learning. The importance-weighted data should be unbiased: that is, the reconstructed distribution should be equal to the original distribution of the original data. If that holds, active learning will never be worse than random sampling, for any classifier. The possible improvement of active learning would lie in its ability to create this 'perfect' importance-weighted selection with fewer queries than random sampling.

Question: Is this true? Does perfect importance weighting equal good reusability?

The interesting element of the IWAL algorithms of Beygelzimer et al. (2009, 2010) and the Vowpal Wabbit is that the importance weight is the inverse of the selection probability. This means that, on average, after selecting many examples, the importance-weighted dataset will be the same as the original distribution of the data.

Question: Do the IWAL algorithms create good importance weights?

There are several reusability problems in active learning that do not occur in IWAL:

1. Areas are of interest to the consumer but not to the selector. Samples from these areas will have a low selection probability. If some examples from this area are selected, they get a high importance weight and everything will be okay. If no examples from this area are selected, it must be a low-density area. Random sampling will have the same problem finding the examples and they will have little effect on the error of the consumer.

2. Somewhat similar, the missed cluster effect. IWAL does not have this problem. It may severely undersample some areas, but it will always sample at least some examples and will give them large importance weights. If it is a high-density area, some examples will eventually be selected and will get a high importance weight. If it is a low-density area, missing the area does not influence the result.

There is a potential problem in IWAL: *Giving large weights to outliers.* Outliers have a small selection probability, but if they are selected they get a very high importance weight, which makes them more important than they actually are. If the number of samples is limited, the variability that is introduced by these very large importance weights might cause problems.

For this to be a problem the outliers should 1. cause a significant change

in the classifier, 2. be rare and outlier enough to not be a problem for random sampling, but 3. still be common enough to be picked by the IWAL algorithm. Requirements 2 and 3 are conflicting: the samples cannot be rare and common at the same time. Perhaps this can be solved by having many *different* outliers: individually they are rare, but there are enough to ensure that often one or two will be selected.

I tried to design a dataset that has these properties (figure 39): a 2D two-class problem, with a high-density cluster of linearly separable data in the middle, surrounded by a low-density circle of outliers. The idea was that there are enough examples in the circle to get a few of them in every selection, but because they are dissimilar they would still get a high importance weight. Random sampling would also select a few of the outliers, but would not give them an unreasonably high importance weight.

I used this dataset with LDA, QDA and SVM consumers, but only with QDA the IWAL selection was worse than random sampling (table 2. The lower the density of the circle, the larger the difference became. For higher-density circles the IWAL algorithm performed reasonably well.

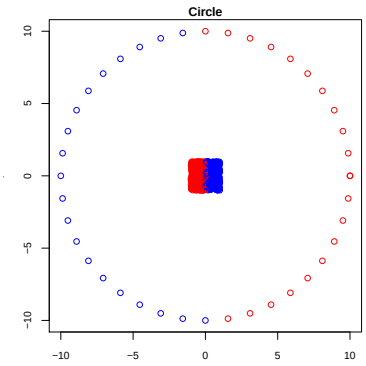


Figure 39: The 'circle' dataset.

# unlab.	IWAL error	Random error	SD of IWAL mean	SD of random mean	P value
10	0.1573237	0.1565162	0.005832228	0.005865937	0.4611302
50	0.05967161	0.06077859	0.002157443	0.00229354	0.6373953
100	0.05496723	0.05203855	0.002032815	0.002114526	0.1590784
500	0.04241635	0.02837003	0.001511856	0.0009738798	4.241171e-15
1000	0.0372564	0.02339909	0.001312881	0.0004984032	1.028748e-22
2500	0.03063014	0.02009514	0.0009925821	0.0003564322	3.303184e-23
5000	0.02538832	0.01889917	0.0008069641	0.0003768349	2.265099e-13
10000	0.02075104	0.01631644	0.0005217249	0.0003384926	6.525832e-13

Table 2: Errors of QDA with IWAL and random sampling, on the circle dataset with circle density 0.001, for different numbers of available (unlabeled) examples. The mean error with IWAL was significantly higher than with random sampling.

Further experiments showed me that QDA has problems on even simpler datasets. A second dataset was a 1D dataset with data on $+1 : (-1, 0)$ and $-1 : (0, 1)$ and a low-density cluster of examples of the +1 at 10. The lower the density of the outlier cluster (e.g., 0.01 or 0.001), the larger the difference between IWAL-QDA and random-QDA.

Even in a simple 1D dataset with linearly separable classes, with the examples distributed uniformly on $(-1, 0)$ and $(0, 1)$, the IWAL selection gave worse results than random sampling (3).

Question: Why does QDA have problems with IWAL-selected examples?

# unlab.	IWAL error	Random error	SD of IWAL mean	SD of random mean	P value
10	0.08630196	0.08639822	0.002804656	0.002761254	0.5097557
50	0.04276652	0.04029243	0.001316769	0.001297164	0.09040261
100	0.03585281	0.0297708	0.001143582	0.0009409249	2.041687e-05
500	0.02495182	0.01569501	0.0009082544	0.0004048721	1.079945e-20
1000	0.02123063	0.01168171	0.0007709762	0.0002928781	9.287979e-31
2500	0.01513897	0.008256677	0.0006385742	0.0001783873	3.706752e-25
5000	0.01066477	0.006336779	0.0004294372	0.0001411677	9.454053e-22
10000	0.008111018	0.00476189	0.0002525977	9.670968e-05	7.837474e-35

Table 3: Errors of QDA with IWAL and random sampling, on the simple 1D dataset, for different numbers of available (unlabeled) examples.

12 On reusability in IWAL

I thought and wrote a bit about reusability of the sample selections that are made by the family of IWAL algorithms.

What is sample reusability?

The definition by Tomanek (2010). It is perhaps useful to note that S_{T_1} and S_{RD} should have the same size.

Definition 12.1 (Sample reusability). Given a random sample S_{RD}, and a sample S_{T_1} obtained with AL and a selector based on learner T_1, and a learner T_2 with $T_2 \neq T_1$. We say that S_{T_1} is reusable by learner T_2 if a model θ' learned by T_2 from this sample, i.e., $T_2(S_{T_1})$, exhibits a better performance on a held-out test set $\mathcal{T}$ than a model θ'' induced by $T_2(S_{RD})$, i.e., $\text{perf}(\theta', \mathcal{T}) > \text{perf}(\theta'', \mathcal{T})$.

Importance weighting

With good importance weights, it is possible to reconstruct the original distribution from a biased (but importance-weighted) sample selection. If the importance weights are more or less correct, the expected error of a classifier trained on the importance-weighted selection S_{IW} is as good as the same classifier trained on a larger random selection S_{RD}. The key problem is how to find the importance weights.

Active learning

Are there universally uninteresting examples? That is, are there samples that no classifier will ever need? This must be untrue. For example, there are density-based classifiers that need all available data.

Therefore, active learning without importance weights can only work if the active learner knows which examples are of interest to the classifier. This makes the sample selection biased towards the selecting classifier. The selection strategy will discard examples that, since there are no universally uninteresting examples, would have been useful to some other classifier. This implies that without importance weighting there can be no universal reusabilitiy, i.e., you can always find a selector-consumer pair where the consumer learns more from random samples than from the samples actively selected for the selector.

Importance-weighted active learning (IWAL) is different. With importance weighting, it is possible to train a classifier on a small importance-weighted sample selection S_{IW} and get the same performance as a classifier trained on a larger random sample selection S_{RD}. (Similarly, for equal-size S_{IW} and S_{RD}, the performance with S_{IW} will be better or equal to the performance with S_{RD}.) The only requirement is that the true distribution can be reconstructed from the importance-weighted set S_{IW} with equal (or greater) accuracy as the reconstruction from the random set S_{RD}. As long as it can keep that promise, the active learning strategy can lower the number of labelled samples in S_{IW} as much as it wants.

IWAL by Beygelzimer et al.; the Vowpal Wabbit

The IWAL algorithms by Beygelzimer et al. (2009, 2010) set the importance weight equal to the inverse of the selection probability (which is always greater than zero). This guarantees that the average importance-weighted sample selection is an unbiased estimator for the true distribution. The importance-weighted selection of a single run with an unlimited supply of unlabelled data will eventually converge to the true distribution. Similarly, the combined sample selections of an infinite number of independent runs with a limited number of unlabelled samples will also converge to the true distribution.

Experiment 1: one run of the algorithm on a very large dataset. The importance-weighted sample selection will have a distribution that is similar to that of the original dataset. Experiment 2: many runs on small random subsets of the dataset. The combined importance-weighted sample selection of all runs will have a similar distribution that is similar to that of the original dataset. Conclusion: the importance-weighted samples selection of the IWAL algorithm is an unbiased estimator of the true distribution.

However, the conclusion that the IWAL algorithm produces unbiased sample selections does not imply that these sample selections are also reusable. Reusability depends on the average performance of the classifier, which must be at least as good as the average performance with random sampling.[5]

On average the IWAL-produced sample selections follow the true distribution of the data. However, the individual runs are not a perfect reconstruction of the true distribution: it is impossible to reconstruct the true distribution from a limited number of samples, except for very special cases. This means that the classifiers that are trained on these individual runs will not know the true distribution and will therefore have an increased error rate. Since the classification errors are all greater than zero, they do not average out: unlike the density, it is not possible to compensate misclassifications by averaging over many runs.

Of course, random sampling has similar problems as IWAL. Like the IWAL selection, a limited random selection can not reconstruct the true distribution. The difference is that the random selection does not have importance weights, it just has too many or too few examples in some areas. An individual selection made by IWAL has similarly undersampled and oversampled areas. In addition, the IWAL selection has importance weights that influence the distribution of the sample selection even further.

[5] While reusability is defined with the expected performance, it may also be interesting to look at the variability of the performance. An active learner that performs slightly worse than random sampling, but does so more reliably may still be of value.

The selection probability in IWAL

In random sampling, the probability that x is in the labelled set is equal to its density: $P(x \in S_{RD}) = P(x)$.[6] In IWAL, the probability that x is in the labelled set depends on two things: the density, i.e., the probability that the sample is offered to the algorithm, and the probability $s(x)$ that the algorithm decides to label the example: $P(x \in S_{IWAL}) = P(x) \cdot s(x)$. The IWAL example receives the importance weight $\frac{1}{s(x)}$, so the expected density of x is the same in both algorithms: $P(x \in S_{RD}) = \frac{1}{s(x)} \cdot P(x \in S_{IWAL})$ and the sample selection is unbiased.

[6] To keep it simple, I left the number of labels out of the probability definitions. I do not think it changes anything important; just imagine that the sets S_{RD} and S_{IWAL} are of the same size (implying the IWAL algorithm must have seen more examples than the random sampler) and that the probabilities are normalised to reflect that.

Although the averaged, importance-weighted density of S_{IWAL} is the same as that of S_{RD} and the true distribution, there is an important difference in the unweighted density. In IWAL, the unweighted probability of seeing x, $P(x \in S_{IWAL})$, depends on the selection probability $s(x)$ that is determined by the algorithm. On average, compared with random sampling and the true distribution, the IWAL sample selection will have relatively more examples for which $s(x)$ is large and relatively fewer examples for which $s(x)$ is small. This is normal, since the IWAL algorithm would not be an active learner if it did not influence the sample selection.

Looking at the IWAL algorithms, we see that it has a preference for examples that are interesting to its classifier: $s(x)$ is larger if example x is interesting and smaller if it is not (figure 40). For example, the AALWC algorithm of Beygelzimer et al. (2010) uses the query probability (the P_k in that paper is the $s(x)$ in the previous discussion):

$$P_k = \min\left\{1, \left(\frac{1}{G_k^2} + \frac{1}{G_k}\right) \cdot \frac{C_0 \log k}{k-1}\right\} \quad \text{where} \quad G_k = \text{err}\left(h'_k, S_k\right) - \text{err}\left(h_k, S_k\right)$$

The choice of $s(x)$ depends on the classifier that is used in the selection. There will be more examples that are interesting to that classifier and fewer of the others, which may be interesting to another classifier. How does this influence the sample reusability?

- In the limit, given unlimited samples, it is not a problem: some of the less interesting examples will be eventually be selected and then the larger importance weights will make everything right. (But as the Martingale gamblers found out: unlimited budgets are scarce.)
- For large sample sizes, there is a problem, though probably not too severe: there will still be enough of the uninteresting samples and with the importance weights most of the structure of the distribution can be recovered. However, there will be much less detail in these areas than in the areas that were interesting to the original classifier. The IWAL algorithm is a bit like a fisheye lens: exaggerated detail in the center and little detail (but a wide view) of the other areas.
- For the smallest sample sizes, problems can be most severe: some of the uninteresting areas are completely empty, those that are selected have an importance weight that is much too large.

Preliminary conclusion: if there is a limited number of samples, there is probably no universal reusability in the IWAL algorithms. It may, however, work fine for many combinations, if there is any correlation in what the classifiers 'like'.

To have universal reusability the selection probability should not depend on a classifier, so maybe an importance-weighted density-sampling algorithm could help there. By undersampling high-density areas such an algorithm may be able to learn the importance-weighted sample selection with fewer samples than random sampling. It will, however, have less detail in the 'important' areas than the classifier-dependent IWAL algorithms, so it will probably not perform better than self-selection IWAL.

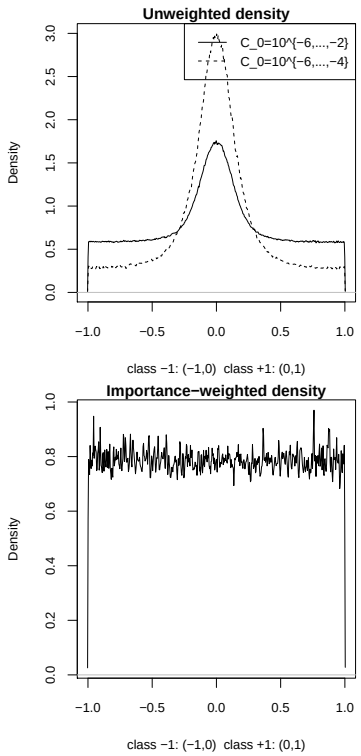


Figure 40: Two plots of the density of IWAL selections made by the Vowpal Wabbit (1000 runs with 1000 unlabelled examples, for various C_0). The 1D problem has two uniform classes, at -1 at $x = [-1, 0)$ and $+1$ at $x = [0, 1]$. The importance-weighted density (bottom) follows the true class distribution, but the unweighted density (top) shows that the algorithm selects most examples from the area around the decision boundary. The spike in the center is more pronounced if the IWAL algorithm is made more aggressive (smaller C_0); most queries outside that area were made with the less aggressive settings (larger C_0).

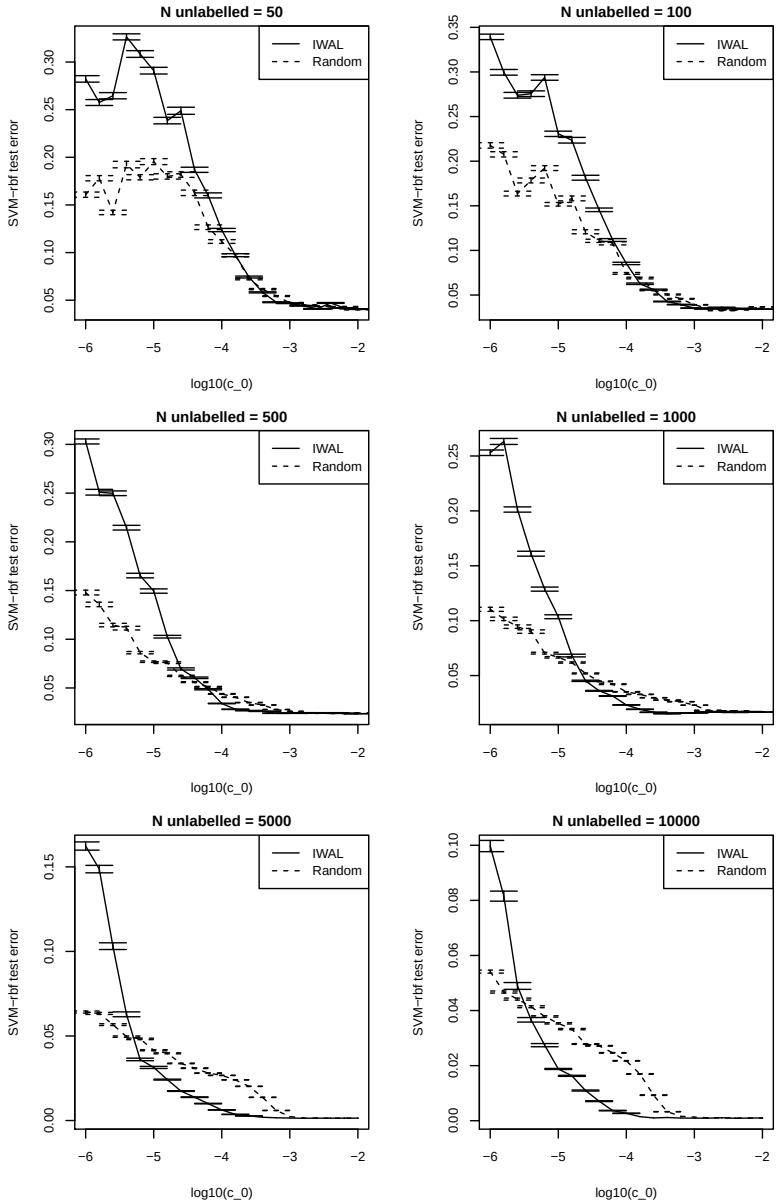


Figure 42: The test error of a radial basis SVM classifier trained on samples selected by the Vowpal Wabbit IWAL algorithm. The dataset is the dataset shown figure 41.

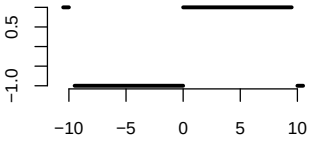


Figure 41: A simple 1D dataset. Most of the samples are in the middle clusters, each of the clusters on the side has 1/92 of the samples.

13 Experiments

Do the theoretical problems discussed in the previous sections also occur in practice? This section presents the results of reusability experiments I did with several classifiers on five datasets, from the UCI Machine Learning Repository and the Active Learning Challenge 2010. I used three selection strategies: random sampling, uncertainty sampling and importance-weighted active learning. Out of curiosity, I also looked at importance-weighted active learning without importance weights – using the same sample selection but with the importance weights set to 1.

The datasets I used are not representative for real-world problems, and neither are these experiments. These experiments do not make any predictions about reusability in practical applications – making such predictions would be very hard, if not impossible. Still, it is useful to do these experiments because they can show that the reusability problems discussed before, and demonstrated with hand-crafted datasets, also occur with independent datasets. The UCI datasets may be unrepresentative and sometimes synthetic, but at least they are not *designed* to cause reusability problems.

Please note that the results I present here only relate to *reusability*. The active learning results on the following pages may seem disappointing, but they do not show the results for self-selection; the results of importance-weighted active learning with self-selection may or may not be different.

In the rest of this section, I first provide more detail about the datasets, the sample selection strategies, the classifiers and the procedure I followed for my experiments. Then I discuss the results, illustrated by the most interesting of the learning curve graphs. The full set of graphs is available at the end.

Datasets

I used five datasets for these experiments. Three datasets come from the UCI Machine Learning Repository (Frank and Asuncion, 2010): `car`, `bank` and `mushroom`. Two other datasets come from the Active Learning Challenge 2010 [7]: `alex`, a synthetic dataset, and `ibn_sina`, from a real-world handwriting recognition problem. Table 4 shows some statistics about these datasets. All datasets are binary classification problems.

[7] `http://www.causality.inf.ethz.ch/activelearning.php`

The car evaluation dataset is a somewhat synthetic dataset. The examples were derived from a hierarchical decision model, so there is exactly one example for each feature combination. This makes it an unrealistic choice for active learning, since active learning tries to exclude examples that are similar and depends on the density distribution of the samples. Another synthetic dataset is Alex, from the Active Learning Challenge 2010, a toy dataset generated with a Bayesian network model for lung cancer.

Finally, it is interesting to mention the Ibn Sina (`ibn_sina`) dataset, a handwriting recognition problem. The objective of this real-world problem is to spot arabic words in ancient manuscripts. The size of this dataset made the experiments with this dataset considerably slower than the other experiments, but it is perhaps the closest to a 'real' active learning problem.

Dataset	Source	Features	Examples	Positive	Test proportion
car	UCI	6, categorical	1728	30.0%	10%
bank	UCI	16, categorical, numerical	4521	11.5%	10%
mushroom	UCI	20, categorical	8124	51.8%	20%
ibn_sina	Active Learning Challenge	92, binary, numerical	20722	37.8%	20%
alex	Active Learning Challenge	11, binary	10000	73.0%	20%

Table 4: Statistics of the test sets used in these experiments.

Sample selection

In these experiments, I compared random sampling, uncertainty sampling and importance-weighted active learning. The only practical implementation of importance-weighted active learning is the Vowpal Wabbit[8], a fast open-source program for online learning. The Vowpal Wabbit creates linear classifiers with or without active learning. I modified the program to have it produce sample selections instead of classifiers. I also added uncertainty sampling as an alternative active learning strategy.

The selector in these experiments is always a linear classifier, which is what the Vowpal Wabbit does. The selection strategy is either random sampling, uncertainty sampling or importance-weighted active learning. For uncertainty sampling it is easy to get a sample selection of a certain size: simply pick the first n samples from the selection. In importance-weighted active learning, the number of samples depends on the C_0 parameter and on chance, but the random component of the selection means that the actual number of samples varies between different iterations (Figure 43).

From the three selection strategies, only importance-weighted active learning uses importance weighting. To determine the effect of the importance weights, I copied the selections from importance-weighted active learning and set the importance weights to 1 for all examples. This selection strategy is listed as 'IWAL (no weights)' in the graphs.

[8] `http://hunch.net/~vw/`

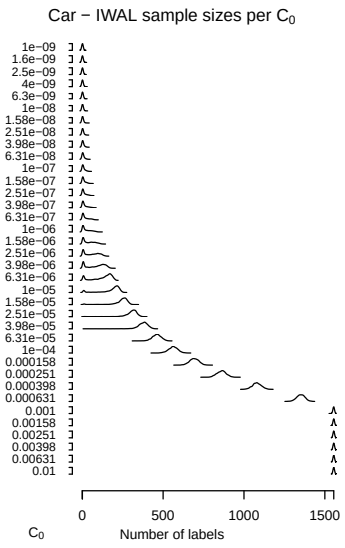


Figure 43: The sample selection size of importance-weighted active learning depends on the C_0 parameter, but it is also a random variable. This graph shows the range of sample sizes for the `car` dataset. I chose the range of C_0 large enough to include both the very small and the very large sample selections.

Classifiers

The selector is always a linear classifier, but I used a larger range of classifiers for the consumer. I experimented with six classifiers, all with support for importance weights:

- Linear regression (`lm`), probably the most similar to the linear model in the Vowpal Wabbit.
- Linear discriminant analysis (`lda`), also a linear classifier but based on different principles.
- Quadratic discriminant analysis (`qda`).
- Support vector machines (`wsvm-linear`), using a third approach to linear classification.
- Support vector machines with a third-degree polynomial kernel (`wsvm-poly3`).
- Support vector machines with a radial-basis kernel (`wsvm-radial`).

Unfortunately, not every classifier worked on every dataset. Especially the LDA and QDA classifiers often complained about singular datasets. In these cases there are no results for these classifier/dataset pairs.

Implementation

I did the experiments with the Vowpal Wabbit for the selection and the R package for the final classifiers. The experiments followed this procedure:

1. For each iteration, before each selection, split the data in a training and test set.

2. For the random selection (Random): shuffle the training set and select the first n samples, for n at fixed intervals from only a few samples to every available sample.
   For the uncertainty selection (US): use Vowpal Wabbit-based uncertainty selection to rank the samples. Select the first n samples, for n from small to large.
   For importance-weighted active learning (IWAL): use the Vowpal Wabbit with C_0 values from 10^{-9} to 10^{-2}. For the datasets in these experiments this range of C_0 produces sample selections with only a few samples, selections that include every available sample and everything in between. For IWAL without weights, use the IWAL selection but with weights set to a constant 1.

3. Train each of the classifiers on each sample selection. For each combination, calculate the error on the test set.

Results

The learning curves for the experiments are shown in the following figures. The line indicates the mean classification error on the test set, the semi-transparent band shows the standard deviation of the mean.

For random sampling and uncertainty sampling these plots are straightforward: I calculate the mean for each value of n. The calculations for importance-weighted active learning are more complicated. Because the number of samples is a random variable, the sample sizes are spread out (Figure 43). There is seldom more than one value at a specific sample size. To get useful data points for these experiments, I grouped the results for the same C_0 and calculated the mean and standard deviation and show them at the median sample size for that C_0.

The learning curves on the Ibn Sina dataset shown in Figure 44 are what you would like to see. The polynomial-kernel support vector machines (left) perform very well if the samples are selected with importance-weighted active learning: the samples are more useful than those of random sampling, so they are certainly reusable. For the radial-basis-kernel importance-weighted active learning does not perform better than random sampling, but it not worse either; uncertainty sampling, however, does not work well.

In other cases the results of importance-weighted active learning is close to that of random sampling (Figure 45), while uncertainty sampling is worse

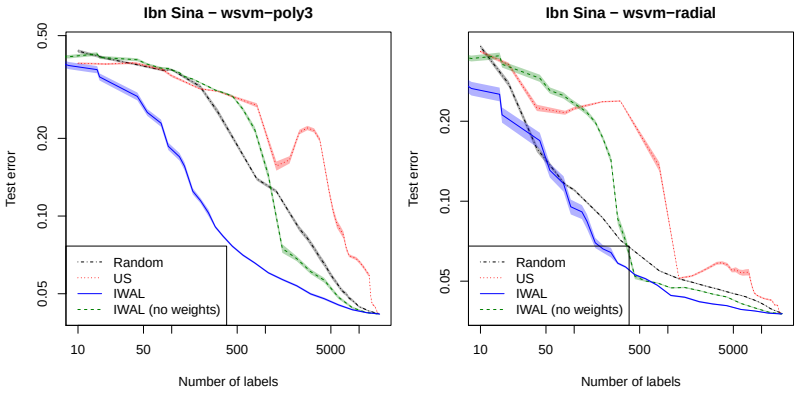


Figure 44: Positive results of importance-weighted active learning on the Ibn Sina dataset with polynomial and radial-basis support vector machines.

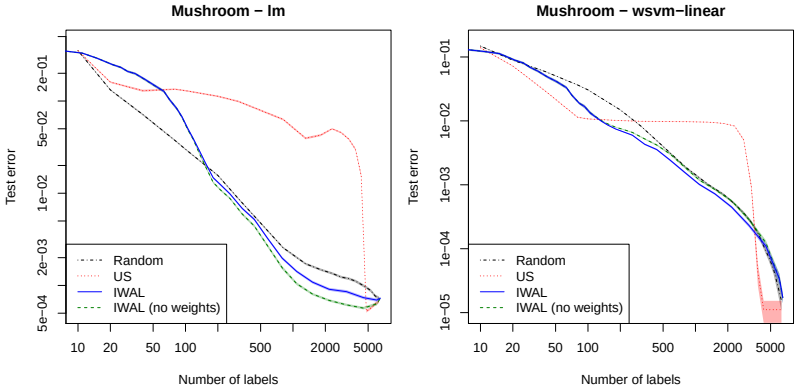


Figure 45: On the mushroom dataset, importance-weighted active learning has an advantage over uncertainty sampling, which may have found a missing cluster. Similar behaviour can be seen in other datasets.

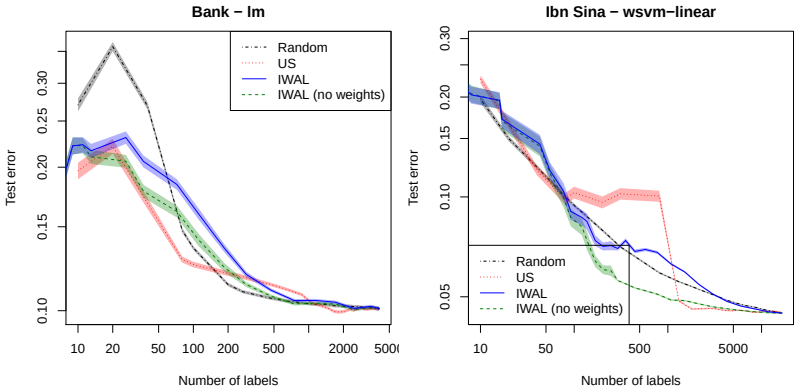


Figure 46: Sometimes importance-weighted active learning is not better than random sampling or uncertainty sampling, but also not much worse.

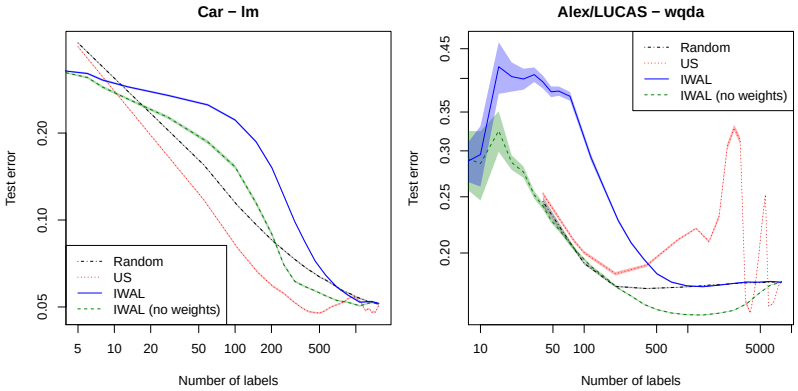


Figure 47: Sometimes importance weighting is the problem: removing the weights from the importance-weighted selection improves the results. This happens most often with LDA/QDA and on the Alex dataset and could be due to the instability that is introduced by the large importance weights.

than random sampling. Sometimes there is not much difference between the sampling strategies (Figure 46). However, importance-weighted active learning can also perform worse than the other sampling methods.

There are even quite a few examples where it helps to remove the importance weights: this often happens with LDA and on the Alex dataset (Figure 47). Perhaps this is a result of the variability from the large importance weights. Even more surprising: in most cases when there is no reusability in importance-weighted active learning, the results of unweighted importance-weighted active learning is still reusable.

From these results it becomes clear that importance-weighted active learning is probably not a solution for the reusability problem. There are certainly cases where the samples are reusable. From the experiments discussed here, one could even get the impression that it is reusable in more cases than uncertainty sampling. But there are too many examples where the selection from importance-weighted active learning is not always reusable.

A second conclusion from these experiments is that importance weighting is not always helpful. The bias correction is correct on average, but in individual sample selections it is imprecise. As predicted in the previous sections, the variability that is introduced by the large importance weights sometimes leads to a performance that is much worse than without the weights.

More results

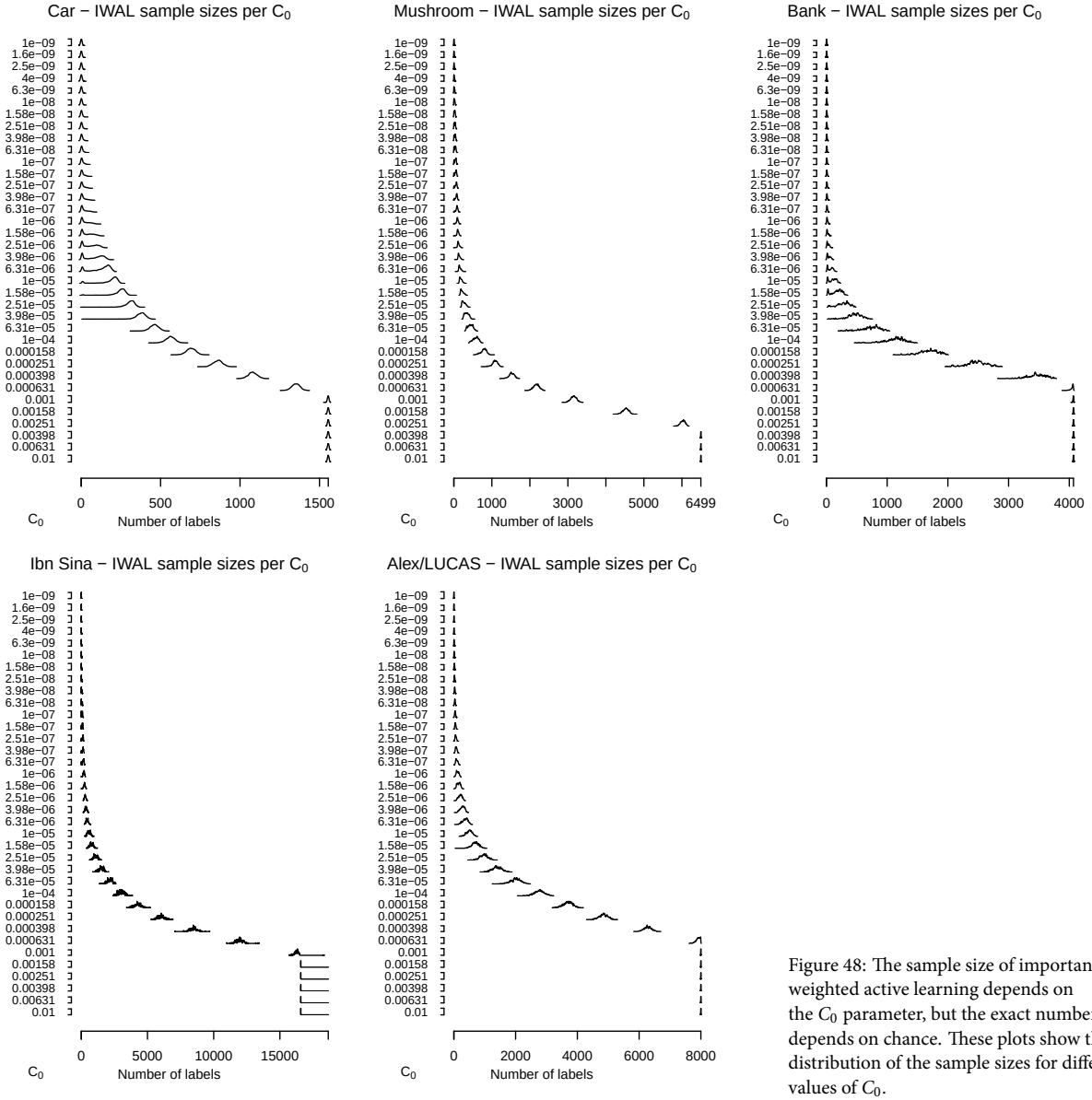
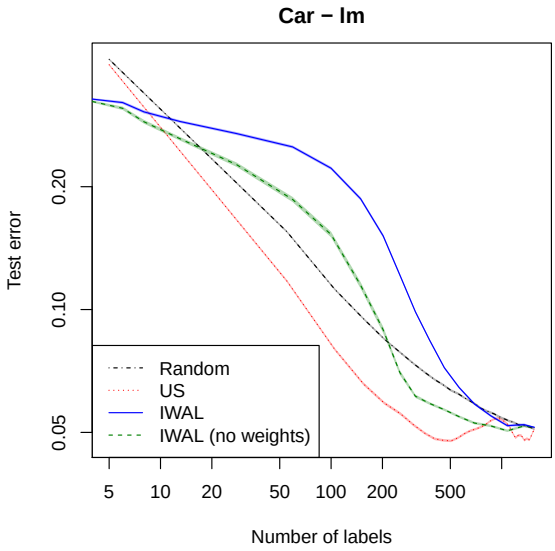
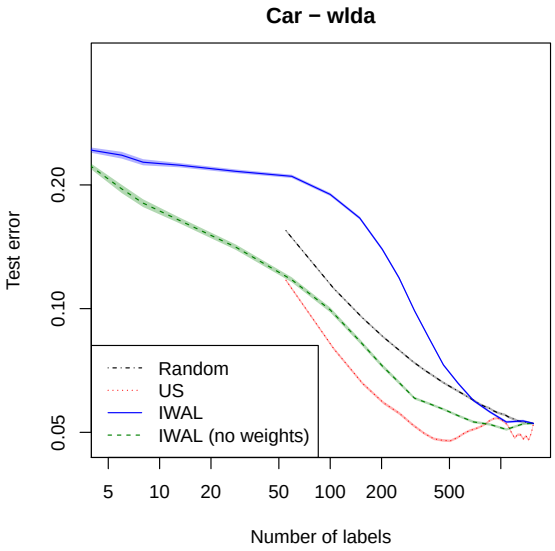
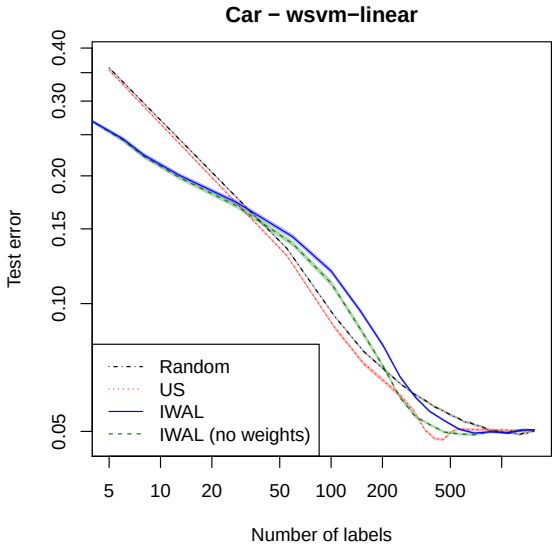
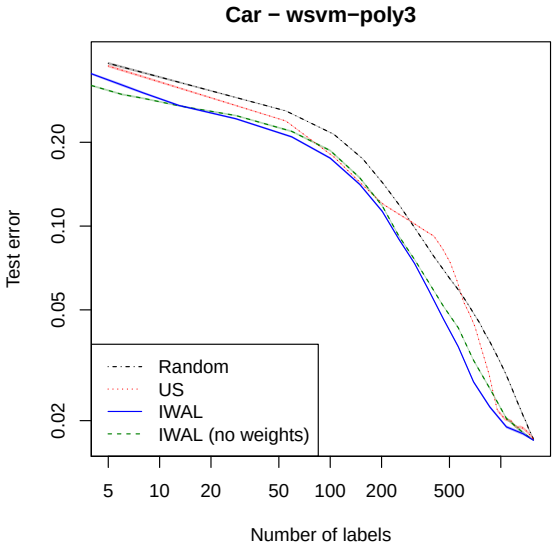


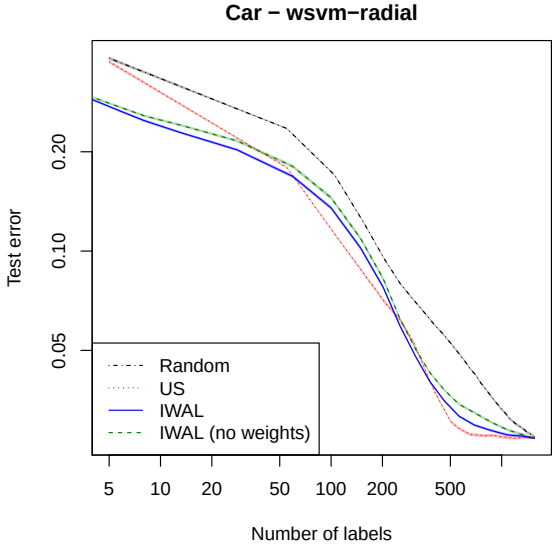
Figure 48: The sample size of importance-weighted active learning depends on the C_0 parameter, but the exact number depends on chance. These plots show the distribution of the sample sizes for different values of C_0.

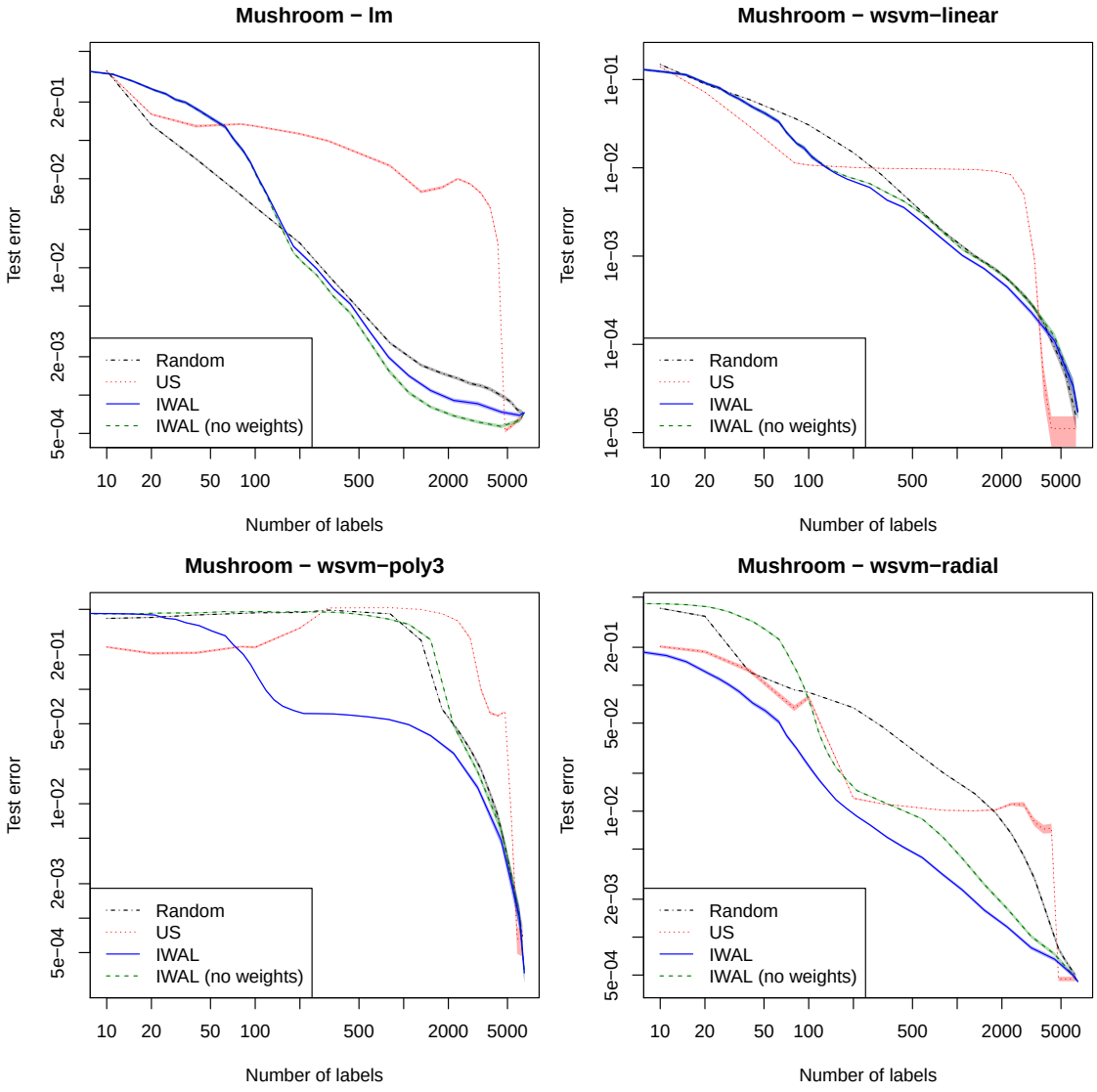

Car − lm
Test error
0.05
0.10
0.20
Random
US
IWAL
IWAL (no weights)
5
10
20
50
100
200
500
Number of labels

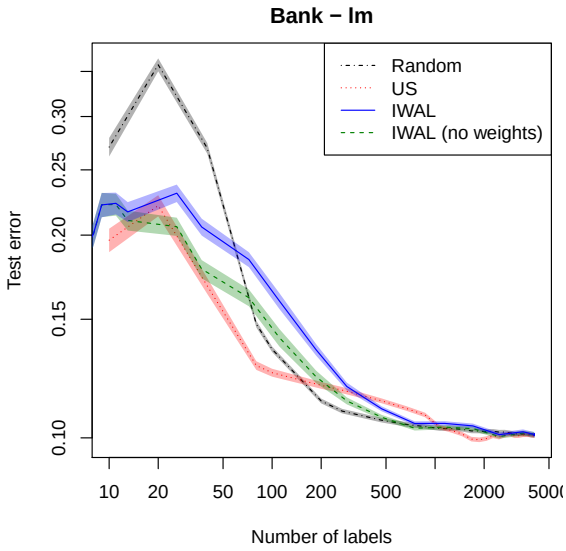

Car − wlda
Test error
0.05
0.10
0.20
Random
US
IWAL
IWAL (no weights)
5
10
20
50
100
200
500
Number of labels

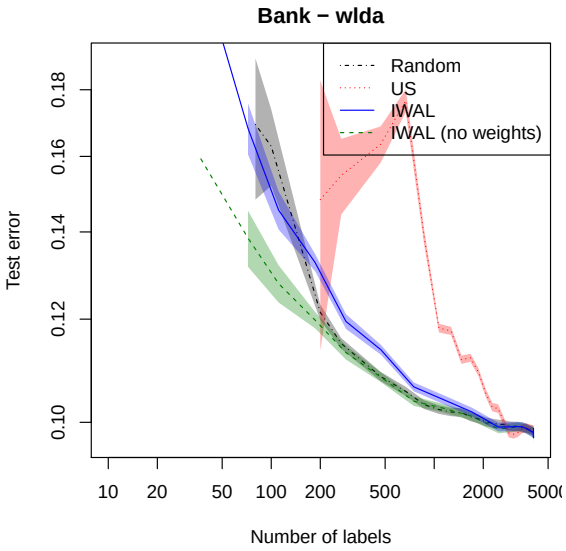

Car − wsvm−linear
Test error
0.05
0.10
0.15
0.20
0.30
0.40
Random
US
IWAL
IWAL (no weights)
5
10
20
50
100
200
500
Number of labels

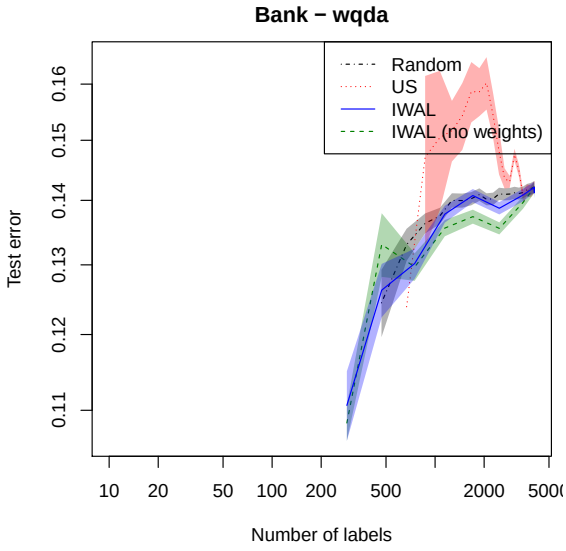

Car − wsvm−poly3
Test error
0.02
0.05
0.10
0.20
Random
US
IWAL
IWAL (no weights)
5
10
20
50
100
200
500
Number of labels

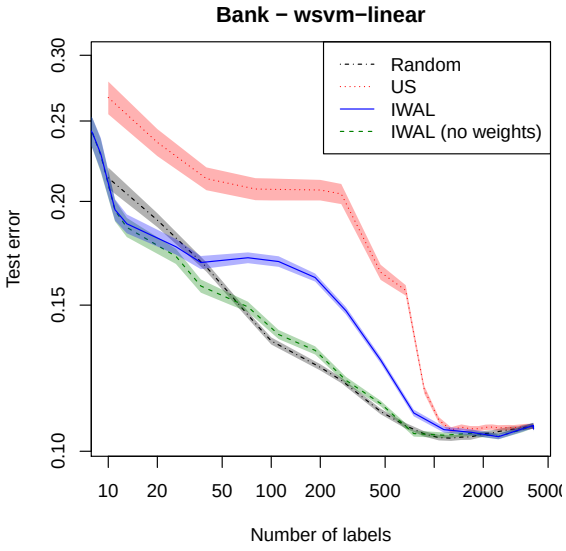

Car − wsvm−radial
Test error
0.05
0.10
0.20
Random
US
IWAL
IWAL (no weights)
5
10
20
50
100
200
500
Number of labels

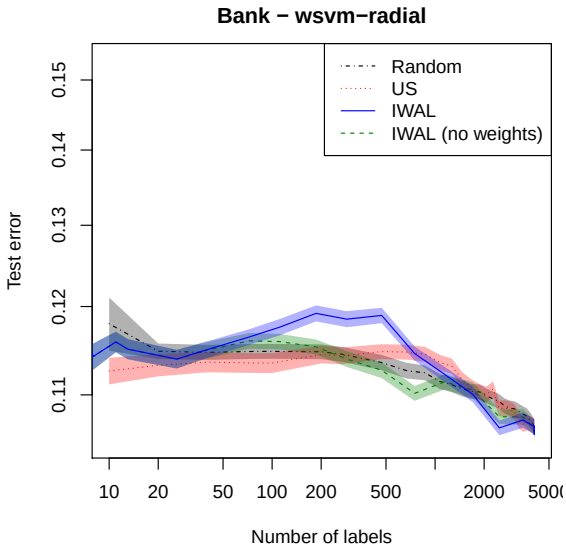

Mushroom − lm
Test error
5e−04
2e−03
1e−02
5e−02
2e−01
Random
US
IWAL
IWAL (no weights)
10
20
50
100
500
2000
5000
Number of labels
Mushroom − wsvm−linear
Test error
1e−05
1e−04
1e−03
1e−02
1e−01
Random
US
IWAL
IWAL (no weights)
10
20
50
100
500
2000
5000
Number of labels
Mushroom − wsvm−poly3
Test error
5e−04
2e−03
1e−02
5e−02
2e−01
Random
US
IWAL
IWAL (no weights)
10
20
50
100
500
2000
5000
Number of labels
Mushroom − wsvm−radial
Test error
5e−04
2e−03
1e−02
5e−02
2e−01
Random
US
IWAL
IWAL (no weights)
10
20
50
100
500
2000
5000
Number of labels

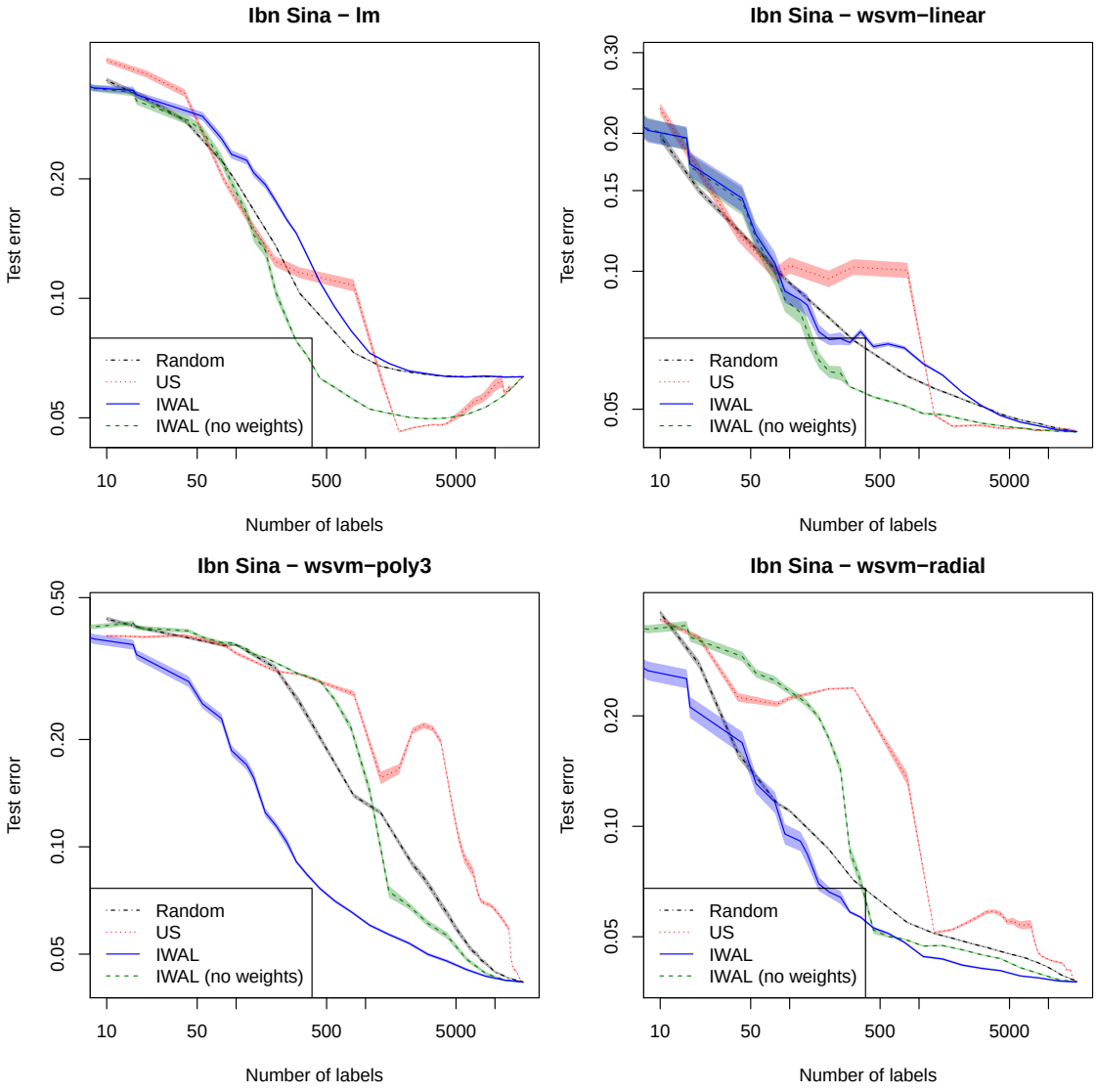

Bank − lm
Random
US
IWAL
IWAL (no weights)
Test error
0.10
0.15
0.20
0.25
0.30
10
20
50
100
200
500
2000
Number of labels

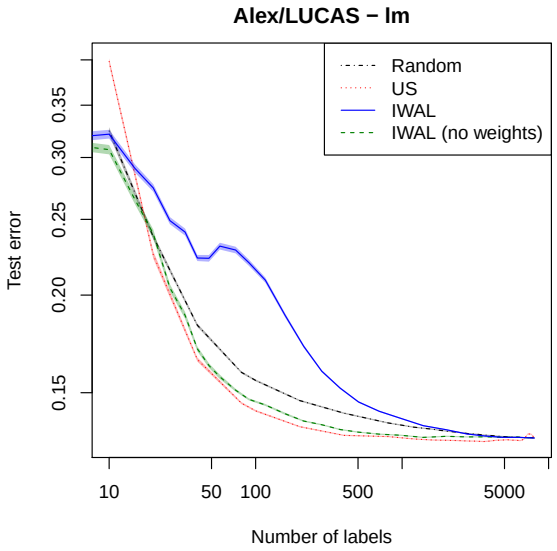

Bank − wlda
Random
US
IWAL
IWAL (no weights)
Test error
0.10
0.12
0.14
0.16
0.18
10
20
50
100
200
500
2000
Number of labels

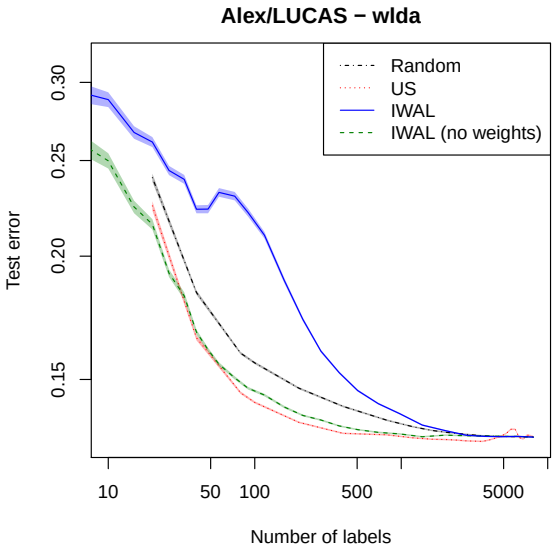

Bank − wqda
Random
US
IWAL
IWAL (no weights)
Test error
0.11
0.12
0.13
0.14
0.15
0.16
10
20
50
100
200
500
2000
Number of labels

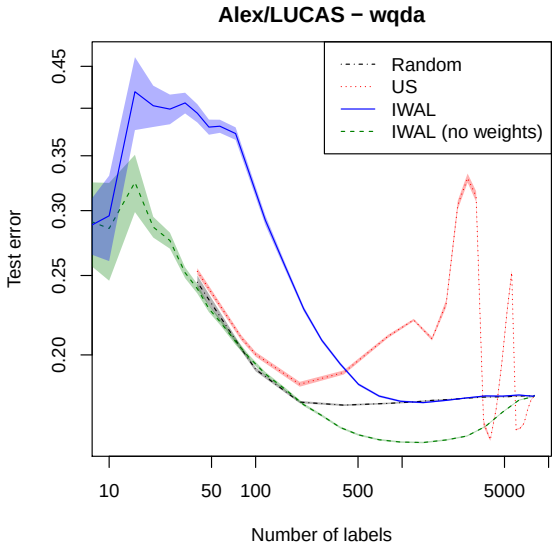

Bank − wsvm−linear
Random
US
IWAL
IWAL (no weights)
Test error
0.10
0.15
0.20
0.25
0.30
10
20
50
100
200
500
2000
Number of labels

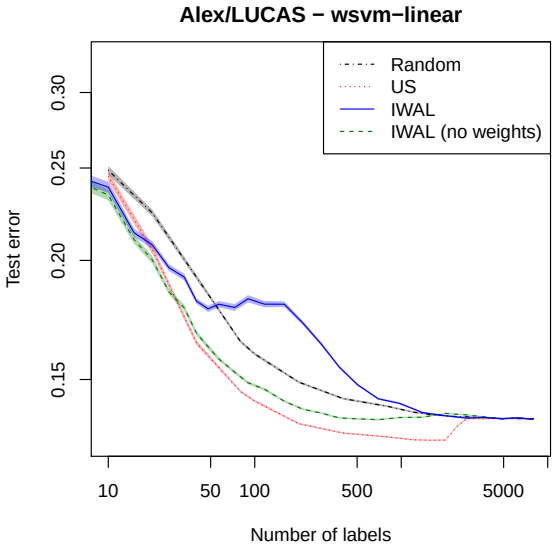

Bank − wsvm−radial
Random
US
IWAL
IWAL (no weights)
Test error
0.11
0.12
0.13
0.14
0.15
10
20
50
100
200
500
2000
Number of labels

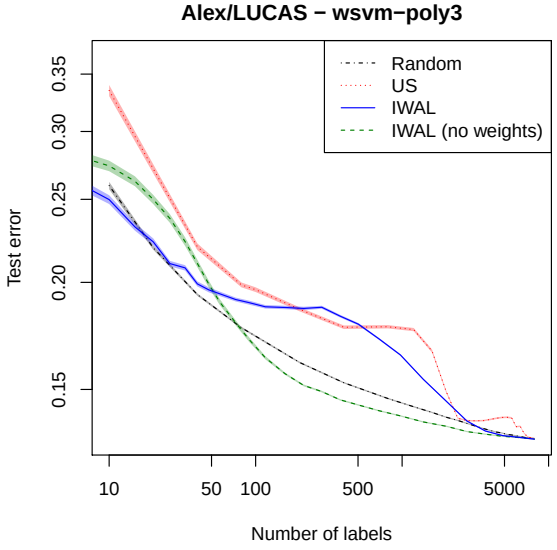

Ibn Sina − lm
Test error
0.05
0.10
0.20
Random
US
IWAL
IWAL (no weights)
10
50
500
5000
Number of labels
Ibn Sina − wsvm−linear
Test error
0.05
0.10
0.15
0.20
0.30
Random
US
IWAL
IWAL (no weights)
10
50
500
5000
Number of labels
Ibn Sina − wsvm−poly3
Test error
0.05
0.10
0.20
0.50
Random
US
IWAL
IWAL (no weights)
10
50
500
5000
Number of labels
Ibn Sina − wsvm−radial
Test error
0.05
0.10
0.20
Random
US
IWAL
IWAL (no weights)
10
50
500
5000
Number of labels

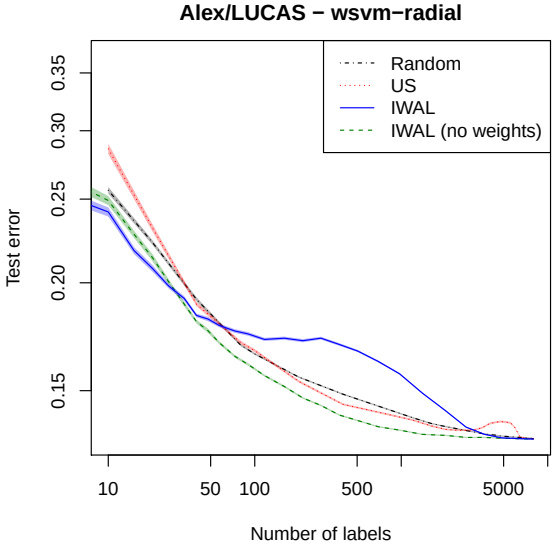

Alex/LUCAS − lm
Random
US
IWAL
IWAL (no weights)
Test error
0.15
0.20
0.25
0.30
0.35
10
50
100
500
5000
Number of labels


Alex/LUCAS − wlda
Random
US
IWAL
IWAL (no weights)
Test error
0.15
0.20
0.25
0.30
10
50
100
500
5000
Number of labels


Alex/LUCAS − wqda
Random
US
IWAL
IWAL (no weights)
Test error
0.20
0.25
0.30
0.35
0.45
10
50
100
500
5000
Number of labels


Alex/LUCAS − wsvm−linear
Random
US
IWAL
IWAL (no weights)
Test error
0.15
0.20
0.25
0.30
10
50
100
500
5000
Number of labels


Alex/LUCAS − wsvm−poly3
Random
US
IWAL
IWAL (no weights)
Test error
0.15
0.20
0.25
0.30
0.35
10
50
100
500
5000
Number of labels


Alex/LUCAS − wsvm−radial
Random
US
IWAL
IWAL (no weights)
Test error
0.15
0.20
0.25
0.30
0.35
10
50
100
500
5000
Number of labels

References

Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance Weighted Active Learning. *Proceedings of the 26th International Conference on Machine Learning*, 2009.

Alina Beygelzimer, John Langford, Daniel Hsu, and Tong Zhang. Agnostic Active Learning Without Constraints. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

Sanjoy Dasgupta. Two faces of active learning. *Theoretical Computer Science*, 412(19):1767–1781, April 2011.

A. Frank and A. Asuncion. UCI Machine Learning Repository, 2010.

Katrin Tomanek. *Resource-Aware Annotation through Active Learning*. PhD thesis, Technische Universität Dortmund, 2010.

Katrin Tomanek and Katharina Morik. Inspecting Sample Reusability for Active Learning. In *JMLR: Workshop and Conference Proceedings 16 (2011): Workshop on Active Learning and Experimental Design*, 2011.